Chao Zhao

Publication Collection

Contents

Publication 1

Constructing a hierarchical user interest structure based on user profiles

Chao Zhao, Min Zhao, and Yi Guan

Constructing a Hierarchical User Interest Structure based on User Profiles

Chao Zhao
School of Computer
Science and Technology
Harbin Institute of Technology
Harbin, China
Email: zhaochaocs@gmail.com

Min Zhao
Baidu Inc.
Beijing, China
Email: zhaomin@baidu.com

Yi Guan
School of Computer
Science and Technology
Harbin Institute of Technology
Harbin, China
Email: guanyi@hit.edu.cn

***Abstract*—The interests of individual Internet users fall into a hierarchical structure which is useful in regards to building personalized searches and recommendations. Most studies on this subject construct the interest hierarchy of a single person from the document perspective. In this study, we constructed the user interest hierarchy via user profiles. We organized 433,397 user interests, referred to here as "attentions", into a user attention network (UAN) from 200 million user profiles; we then applied the Louvain algorithm to detect hierarchical clusters in these attentions. Finally, a 26-level hierarchy with 34,676 clusters was obtained. We found that these attention clusters were aggregated according to certain topics as opposed to the hyponymy-relation based conceptual ontologies. The topics can be entities or concepts, and the relations were not restrained by hyponymy. The concept relativity encapsulated in the user's interest can be captured by labeling the attention clusters with corresponding concepts.**

I. Introduction

Designing systems to retrieve or recommend relevant information to meet users' needs is an increasingly challenging endeavor as the massive stores of online data continue to grow. The most common approach to doing so is based on personalized information related to user interests. These individual interests, which we refer to here as *attentions*[1], are stored in the user profiles. They are a key component in the filtering and recommendation systems of today's web services, e.g., search engines and feeds[1], [2].

User attentions can be built into a hierarchy, where the general and specific interests of users are effectively organized in different levels. According to [3], higher-level interest categories reflect longer-term user interests; lower-level categories reflect the user's current interests. [4] attributes higher-level attentions to implicit, passive interests, while lower-level attentions correspond to explicit, active interests. In any case, this kind of user profile can be effectually utilized for personalization (e.g., query disambiguation, interest expansion, and cold-start problems alleviation) by capturing the semantic or knowledge-level similarities of attentions [5], [6], [7], [8], [9].

Figure 1 shows an example of "Sports-Basketball-NBA-Rockets" as a certain path in the interest hierarchy. Upon capturing these concepts in the user profile, the web search engine is more likely to ascribe the query to the Houston Rockets basketball team rather than spacecraft or other technically unrelated topics when the user simply searches the word "rockets".

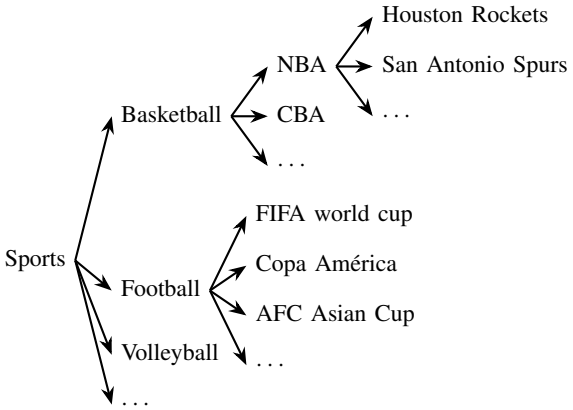


Fig. 1: Example interest hierarchy.

There are many existing methods for extracting hierarchical user interests from the user's behavior (e.g., clicks on or scrolls through various documents or sites). Previous studies have mainly focused on mining the interests of only one individual at a time; research on the inner structure or relativity among these interests themselves is relatively scant. For example, is this hierarchy similar to that of a conceptual ontology? What broader subjects will capture a user's interest, say, if he or she clicks on articles about artificial intelligence (AI)? How similar are this person's interests to another person who is interested in machine learning?

In this study, we attempted to construct an attention hierarchy directly from a set of user profiles, rather than a single profile, using a clustering technique. We explored the factors responsible for aggregating the attentions together into the hierarchy. Our primary contributions are two-fold:

- We organized all the attentions as a user attention network (UAN) based on their co-occurrences in user profiles,

[1]For clarity, we use *attentions* to refer to words representing user interests, e.g., "Harry Potter" or "Fantasy movie".

then divided the UAN into communities constituting an attention hierarchy structure.
- We analyzed the factors responsible for clustering certain attentions according to the attention hierarchy, and found that topics – not concepts – were paramount. Concepts only occasionally play a role similar to topics.

The remainder of this paper is structured as follows. In Section II, we give a brief review of related works on the hierarchical representation of user interests. In Section III, we describe the UAN construction and corresponding community detection method in detail. We present the community detection results in Section IV and discuss the aggregation mechanics of attentions in Section V. A brief conclusion and discussion on future research directions are presented in Section VI.

II. Related works

The extant research on constructing user interest hierarchies can be split into two main categories: Concept-based and content-based.

Concept-based methods model the hierarchy of interests with the aid of concept taxonomies. These include the Open Directory Project[10], [11], [6] or Wikipedia folksonomy[12], [13]. [10] used the first three levels of ODP categories to construct a general interest profile for mapping user queries to a set of categories. [11] mapped each interest of one user as a hierarchical ODP path, then calculated the similarity between the search results and the user profiles and re-ranked the results accordingly for the sake of personalization. [12] mined Twitter user's interests from the entities of their Tweets by finding corresponding concepts in Wikipedia folksonomy. By leveraging the Wikipedia category graph, [13] further constructed interest hierarchies of Twitter users via spreading activation.

Content-based methods involve constructing interest hierarchies without the aid of predefined taxonomy. [14], [15] mined concept-based user profiles from clickthrough data to infer user concept preferences, assuming that general terms with higher frequency would be placed at higher levels in the user profile hierarchy while specific terms with lower frequency were placed at lower levels. They defined two types of relationships, similarity and parent-child, for the concepts c_1 and c_2 in the search results and utilized the similarity measure $Sim(c_1, c_2)$ and conditional probability $P(c_1|c_2)$ and two thresholds to justify the existence of relationships between concepts. [4] extracted the terms from web pages bookmarked by users as their interests, and utilized a divisive clustering algorithm to group these interests into a hierarchy. The correlations between any two terms were measured by the Augmented Expected Mutual Information (AEMI) and the MaxChildren threshold-finding method was used to find a reasonable threshold for correlations. [16] identified mismatching between topic hierarchies used in analytics and learned from log data, which they resolved by first aggregating search queries and clicks into concepts in the hierarchy, then mapping the concepts to a topic taxonomy.

III. Methods

A. User profile

We randomly selected 200 million user profiles from the Baidu search engine, which were obtained automatically from user behaviors (e.g., user search and click histories), to construct our dataset. Each user is associated with a series of attentions given personalized weights. The weight of each attention is an integer between 1 to 2,000 that indicates the importance of this attention to the user. Most of the attentions are entities, such as the "Harry Potter", "Stephen Curry", or "Fan Bingbing"; several were also be concepts, e.g., "fantasy novel", "NBA", or "film star". To improve the quality of the dataset, we only reserved the core attentions of each user with weights of exactly 2,000. We also discarded any users with less than 10 core attentions, because we prefer to believe that these attentions co-occur randomly rather than due to non-trivial factors, e.g., higher-level interests of users.

We obtained a total of 433,379 unique attentions after filtering. We analyzed the distribution of the number of attentions for an individual user as shown in Figure 2: One user can have no more than 250 core attentions. The distribution is power-law like, and most of the users have less than 50 attentions. We also identified the distribution of the number of followers for a unique attention, as shown in Figure 3. Both axes are logarithmic-scaled and explicitly follow a power-law distribution. There are a few attentions that possess a very large amount of followers, while most attentions are associated with relatively few people. In effect, attentions are very personalized attributes for individual users. The most popular concept and entity attentions are "society" and "Fan Bingbing", respectively.

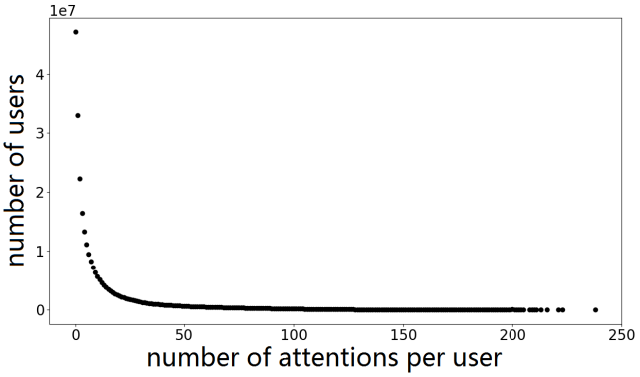


Fig. 2: Distribution of number of attentions in an individual user profile. x-axis represents the number of attentions in the user profile; y-axis represents corresponding user numbers.

B. User attention network

Attentions in one user profile can be used to construct the interest hierarchy for the current user, but are less helpful as far as determining the more general associations among attentions themselves. To capture these associations, we assumed that

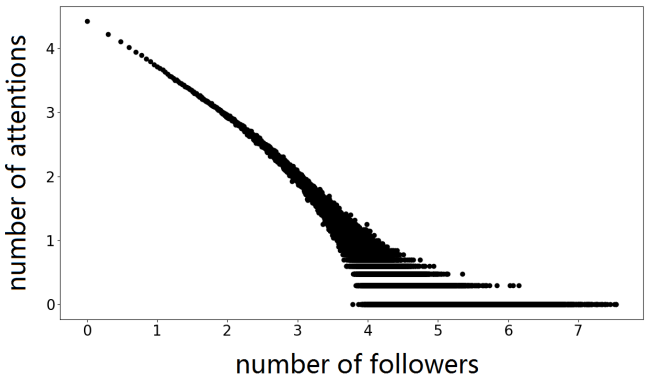


Fig. 3: The distribution of the number of users for a unique attention. X-axis represents the number of followers of an attention, and y-axis represents the corresponding attention numbers. Both two axes are logarithmic scaled.

two attentions in one user profile are related: The more frequently the two attentions co-occur in one user profile, the more relevant the two attentions are.

Based on this assumption, we first constructed a complete, undirected graph for each user profile in which nodes are attentions and the weight of each edge is assigned as 1. We then merged all these graphs into a complex network called the user attention network (UAN). The weight of the same edge was cumulated during the merge process, and finally represented the number of users associated with both attentions. This weight represents the strength of association between the two attentions to some degree, but the association strength is not necessarily strictly proportional to the weight; it also depends on the user count of each attention. For example, there are 300 users with both attentions "J. k. Rowling" and "Harry Potter", but 20,000 users are interested in both "Harry Potter" and "society". In fact, the count of followers of "society" exceeded 3 million with 20,000 co-occurrences, and therefore is trivial in comparison. To get a more reasonable weight, attentions with large user count need to be penalized.

To ensure the edge weights of UAN can better depict the association strength of attentions, we re-calculated the weight of two attentions a_i and a_j as follows:

$$w(a_i, a_j) = \log(1 + w'_{ij}) \times \log(\frac{N}{UC_i + UC_j})$$

where w'_{ij} is the original weight, and UC_i and UC_j are user counts of two attentions. N is a large integer to make sure that the second term is positive, which we arbitrarily set as the number of users in our dataset. The first item of this formula decreases the impact of weight from linear to log-linear, while the second penalizes the weight according to the user count of two attentions.

Table I shows the five nearest attentions with "Harry Potter" before and after the weight re-calculation. The re-calculated results perform much better than the original weights.

TABLE I: The comparison five nearest neighbors of "Harry Potter" before and after weight re-calculation.

before		After	
neighbor	weight	neighbor	weight
society	20052	Daniel Radcliffe	141.774
Qiao Renliang	12606	Hermione Granger	140.087
Love O2O	11898	Emma Watson	138.945
entertainment	11612	Fantastic Beasts	135.415
doctor	10892	Harry Potter 5	134.320

C. Attention community detection

From a complex network perspective, the UAN has a community-based structure. Like Figure 4 shows, several attentions in the UAN are connected more densely than others. These densely connected clusters (communities) of attentions implicitly indicate that they are more likely to appear simultaneously in a single user profile, from which the structure of the attention set can be inferred.

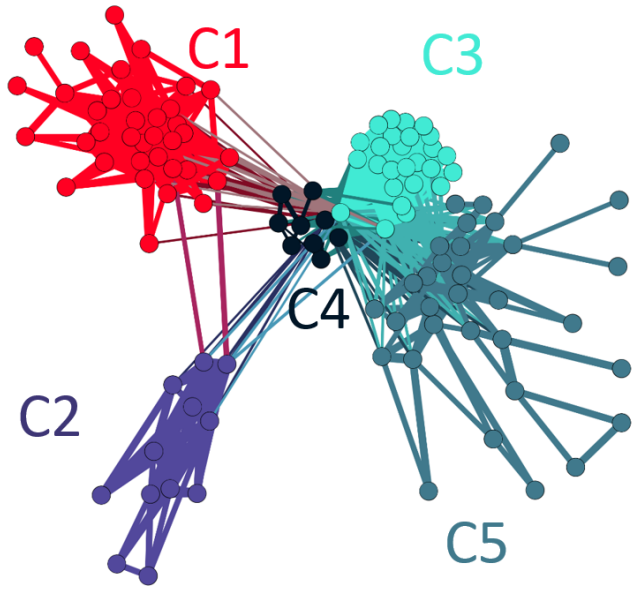


Fig. 4: An toy example of UAN.

The Louvain method (LM) [17] is one of the most widely used algorithm to discover hierarchical communities of nodes from large-scale weighted graphs. Compared with other community detecting algorithms, LM is time-efficient and easy-to-implement. Therefore, we apply it here to identify attention clusters in the UAN. It determines the community label of each node in an iteratively, greedy manner by maximizing a modularity measure Q[18]. Given a partition result, the modularity measures the difference between the edge density

in the inner community and that in a random network:

$$Q = \frac{1}{2m}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m})\delta(c_i, c_j)$$

where c_i denotes the community of node i and $\delta(i, j)$ is the Kronecker delta. A is the adjacency matrix of the network. m is the total number of edges. $\frac{1}{2}\sum_{ij} A_{ij}\delta(c_i, c_j)$ counts all the edges within the same community, and $\frac{1}{2}\sum_{ij} \frac{k_i k_j}{2m}\delta(c_i, c_j)$ is the expected number of edges of these communities in a random network. Q is obtained by normalizing the difference of this two with the number of edges m. Higher modularity means that there are more edges in the inner of community than that expected by chance, and further indicates the existence of potential community structures in the current partition result.

LM first assigns individual communities to each node, then runs two phases iteratively to maximize the modularity. In the first phase, it repeatedly and sequentially merges each node i to the community of one of its neighbors to ensure the maximum increase in modularity until modularity can no longer be improved by an individual move. Based on this result, the second phase regards the detected communities as new nodes to generate a new graph to which the first phase is reapplied. Once the maximum modularity is attained, the algorithm outputs a hierarchical clustering result.

Because the LM uses a heuristic method to determine the community assignment of each node, the result may not be optimal and the divided subgraphs may be still too large. To obtain smaller communities, we ran the Louvain method on the large results iteratively until two terminate conditions were reached:

- The current graph's node size is less than 1,000;
- The current graph's largest detected community covers more than 80% of the nodes.

The hierarchical structures of the results are reserved after each iteration, ultimately yielding an attention hierarchy in which all intermediate nodes and leaves can be regarded as attention clusters. (Leaves are the clusters with minimal sizes.)

D. Community labeling

Attentions in the same communities of the hierarchy co-occur more frequently than those belonging to other communities. In other words, if the user expresses the interest within a certain community, he or she is likely to be interested in other attentions in the same community. By analyzing the common characteristics of attentions in the same hierarchy, we expected to find the underlying reasons that certain attentions were aggregated. It is cumbersome to analyze and summarize these patterns manually, however, since the counts of both attentions and communities are large.

To capture the more general and informative relations, we labeled each community with their corresponding concepts using the Baidu knowledge base (BKB), which provides the hypernym path of one attention. The BKB is a directed acyclic graph (DAG) and contains 13 domains (children) under the root node. For each domain, we used the lowest common ancestor (LCA) concept node of these attentions as labels to summarize the attentions in current community.

Cluster-based results inevitably contain noise, and the BKB may also not return the most appropriate fine-grained hypernym concepts. Both factors can lead to a strict LCA search returning coarse concepts. To avoid this, we relaxed the restrains to find only LCAs covering more than 50% of the attentions of the same domain in one community.

Figure 5 shows a mini version of the clusters containing seven attentions from two domains. The concept "war movie" contains 75% of the attentions in the *Culture* domain, which is marked with green blocks, while the concept "battle" contains all attentions in the *Life* domain marked in yellow. We labeled this cluster as "war movie & battle".

If more than one concept is labeled for a single community, we assume these two concepts have certain associations in regards to user interest level. For example, individuals who are interested in battles may also like films with battle scenes. Not only the leaf communities of the hierarchy, but all the intermediate communities can be effectively labeled with concepts. This can find the relations in a higher level, and avoid the ignorance of interesting concept relations due to the small size of clusters. Generally speaking, concept associations in smaller-size communities are stronger than those in the larger communities.

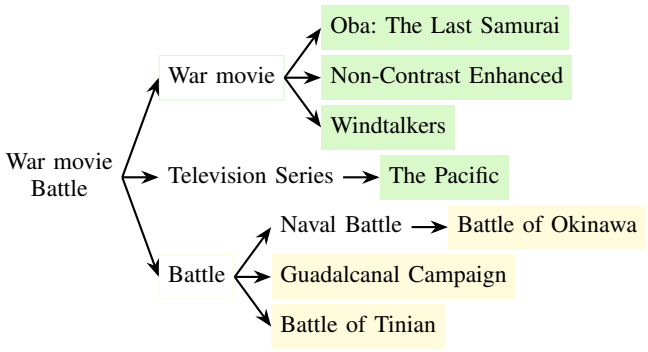


Fig. 5: Example of LCA labeling

IV. Results

After the community detection, we obtained an attention hierarchy which contained 34,676 small communities. Figure 6 shows the distribution of the number of communities and attentions in each level of the hierarchy. The shallowest leaf communities exist in the fourth level of the hierarchy, and the depth of the hierarchy is 26. Most of the communities and attentions are concentrated in the area from the 7th to the 19th level.

The distribution of community sizes is shown in Figure 7. Most of the communities have less then 30 attentions, and the largest size is no more than 1,000 under our partition strategy. Table II shows several communities from the attention hierarchy, which are just those have shown in Figure 4.

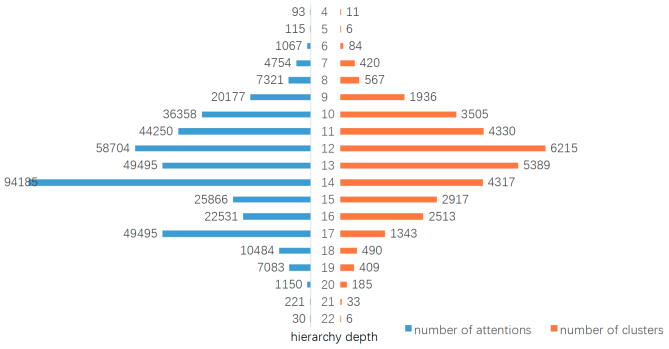


Fig. 6: Distribution of number of communities and attentions over the hierarchy depth.

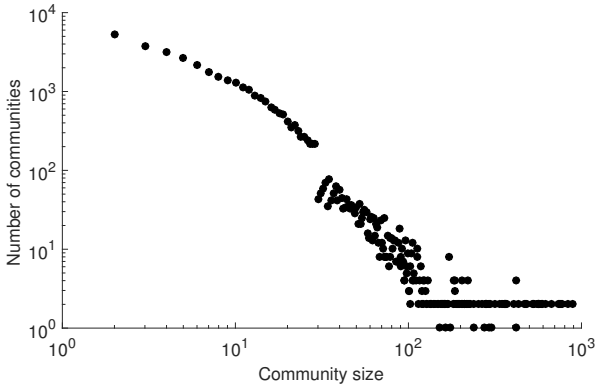


Fig. 7: Distribution of communities by size. X-axis represents the size of a single community; y-axis represents the corresponding community numbers. Both axes are logarithmic-scaled.

Unfortunately, it is difficult to measure the quality of our obtained communities. There are two kinds of indexes to evaluate a clustering result: internal or external. The former needs a pre-defined distance measure between two instances. Once we determine the weights of edges in UAN, the Louvian method can derive a reasonable clustering result accordingly. The latter index, as the name suggests, needs an external reference model (e.g., the clustering result given by experts) to compare with. However, there is no such a golden-standard model to tell us what the clusters of user attention should look like. We therefore have to evaluate the clusters manually, based on the common knowledge of humans. We randomly selected 200 communities from the result of LM, and then to label each as +1 (if more than half of the instances in the community belonged to the same topic or the relations among the topics are easily to understand) or -1 (other conditions). 84% of the sampled communities were labeled as positive, and indicated the reasonability of the our obtained communities to some degree.

TABLE II: Example clusters in table-structured hierarchy.

C1	C1-1	Sirius Black, Hedwig, Quirrell, Grindelwald, Hermione Granger, Luna, Snape, Moaning Myrtle
		Quidditch, Ministry of magic, Felix Felicis, Patronus Charm, Expecto patronum, Broomsticks
	Ravenclaw, Hufflepuff, Gryffindor, sorting hat, Duemstrang Institute, Beauxbatons Academy of Magic, Butterbeer	
	Harry Potter, Harry Potter 5, Harry Potter cast, Daniel Radcliffe, Emma Watson, Bonnie Wright, Matthew Lewis, Michael Gambon, Alan Rickman, Tom Felton	
	house-elf, Voldemort cat, Dementor, Death Eater	
	James Porter, Viktor Krum, Alastor Moody, Weasley, Fleur delacour, Cedric Diggory, Neville Longbottom, Ribus Hagrid, Teddy Lupin, Frank Dillane	
C2	Elijah Wood, Richard Armitage, Miranda Otto, Dominic Monaghan, Viggo Mortensen, Arwen	
	Elrond, Haldir, Earendil, Gil-galad, Glorfindel, Legolas Greenleaf	
C3	artificial intelligence, cloud computing, silicon valley, YAHOO, unmanned driving, Internet + medicine, face recognition	
	Liu Zhen, Ren Zhengfei, Lei Jun, Jack Ma, Rothschild, Wanda, Rupert Hoogewerf	
	BAT, Alibaba stock, Google stock, Apple share price, Tencent market cap, Jingdong stock, Partner system, world's billionaires list	
	Internet e-commerce, Alipay, electronic business platform, WeChat payment, Micro payment, Maker, Self media	
C4	Bodybuilding, McCarthyism, little pink, Zhonghua Jia, Magician, Round table, political correct, empathy, Q&A Platform, sinovation ventures, Carpe Diem	
C5	machine learning, support vector machine, ID3, deep learning, decision tree	

V. Discussion

Compared to the hierarchy structure of conceptual ontologies, in which concepts and entities are organized based on strict hyponymic relations, attentions in our hierarchy are organized by topic.

A. The topic-based organizations

Attention organizations seem intuitively linked to hyponymic relations. For example, a person who is interested in the movie "Harry Potter" would more likely to be interested in other fantasy movies like "Lord of the Rings". According to our community result, however, none of other movies existed

in the leaf community or siblings of "Harry Potter". These attentions instead centered around items like "Emma Watson", "Lord Voldemort", and "Quiddich". They are different concepts — actors, characters, and plot points of the film — but are organized under the same topic "Harry Potter". The community of "Lord of the Rings" also shows similar patterns comprised of actors in the film and its characters. "Harry Potter" or "Lord of the Rings" act as topics responsible for aggregating certain attentions together. They themselves are aggregated by more general topics, e.g., fantasy movies.

The biggest difference between topics and concepts is that concepts have strict top-down structures, are organized by single hyponymic relations, and follow the logical relation of transitivity. "Battle of Okinawa" is a naval battle and thus can falls under the larger "battle" category. The topic, however, is a less strict summary of a local attention set. It does not have to possess transitivity and the relations between attentions may vary. For example, "Emma Watson" falls under the topic "Harry Potter" but not "fantasy movie". The top-down structure is also not abstract. Sometimes a movie can be a topic and aggregate its cast members together, while actors can also form topics into which their films are aggregated together.

Further analysis of other communities indicates that such topic-based aggregation is common. A concept can sometimes play the role of a topic, but not always. Enriching the user's attention via general concepts is not always effective, as discussed above; further, enriching the user's attention to special cases at the concept level may also be ineffective. Consider again our questions posed at the start of Section I regarding interests shared by two users. If we know one person is interested in AI and another is interested in machine learning, the communities encapsulating these two attentions are as shown in C3 and C5 of Table II. The community containing "machine learning" falls under the topic "machine learning techniques", while the community containing AI and its near communities fall under the topic "Internet", which centers more around Internet companies as well as their founders, new technological products, and economic impacts.

These differences altogether demonstrate that the hierarchical structure of our user attentions is not well consistent with that of a conceptual ontology. Though "machine learning" is a hyponym of "AI" in conceptual ontology, most non-experts are disinterested in its concrete algorithms. On the other hand, "Harry Potter" and "Emma Watson", which fall into the same communities, have a low similarity in conceptual ontology as shown in Table III.

TABLE III: Hypernym paths of "Emma Watson" and "Harry Potter".

attention	Hypernym path
Emma Wason	Emma Watson → actor → entertainer → person
Harry Potter	Harry Potter → Fantasy film → film → film and television work → video work → work

B. Topic categories

There are a variety of entities or concepts that may play the role of topics. In the example "Harry Potter", the topic is a movie. The actors or even the director may also be the topic responsible for aggregating certain attentions related to the film together, however. These topics are mainly entities, and the corresponding relationships are the attributes of the topics. Concept-based topics, on the other hand, may aggregate their hyponyms as well as other concepts sharing certain similarities, like the "war film" and "battle" example described above.

Another interest community is shown in C4 of Table II in which there do not seem to be any strong relationships among entities or concepts. The longtime users of Zhihu[2], however, would notice immediately that this cluster contains hot topics in the Zhihu community. In this case, issues of interest within a certain web community play the role of the topic.

The factors that define an entity or concept as the "topic" are still unknown. The category of a given concept itself is apparently not a factor. Though "Harry Potter" and "Oba: the Last Samurai" are both movies, for example, they play different roles in their respective communities.

C. Adding topic relationships to conceptual ontology

Although the conceptual ontologies are not completely consistent with the interest hierarchies, they are still valuable resources in determining the similarities of two entities at the semantic or knowledge level. [3] used spreading activation to incrementally update the interest scores of concepts in user profiles, and then to calculate the similarities between user profile pairs based on the Euclidean distance of concept vectors. [19] applied the random walk algorithm to expand user profiles on the Wikipedia graph. They attempted to use inner Wikipedia links to alleviate the inconsistency between taxonomy and interest hierarchy, but inner links, arguably, fail to reflect entity relations across the user interest perspective.

To bring entities in the same topic closer together in the concept hierarchy, we tried to introduce the "related_to" relationships into the concept hierarchy based on the concept labels discussed in Section III-D. As shown in Figure 5, by adding "related_to" between the concepts "war movie" and "battle", we were able to reflect the relativity of these two concepts at the user interest level. Some other examples are shown at Table IV. Many network-structure based methods (e.g., spreading activation, random walking) can benefit from introducing new relations during the similarity calculation.

VI. Conclusion

In this study, we explored the organization structure of user attentions (interests) from user profiles aggregated via a clustering method. We linked all attentions in user profiles

[2]A Quora-like QA website in China

TABLE IV: Example concepts labeled to the same cluster.

war, war movie, troop, empire, militarist, tactic, arm
athlete, sport competition, sport award, competition schedule, sports channel, live platform, sport lottery
university, major, university rankings, recruitment, recruitment website
plane ticket, flight, hotel, travel agency, visa, tariff
estate, house price, building materials, building design, second-hand house

into a user attention network (UAN), and utilized the Louvain method to obtain the hierarchical attention clusters from the UAN. We found that the user attention hierarchy is mainly organized by the topics, as opposed to hyponymy-relation-based conceptual hierarchies. Topics can be entities or concepts. The factors responsible for aggregation can be attribute relations, hyponymy relations, or more general similarities. Through the clustering results of user attentions, we were able to supply "related_to" relations to the conceptual ontologies to better depict the concept relativities across the user interest perspective.

In the future, we plan to explore the factors that define an entity or concept as a topic. We also will experimentally demonstrate the benefit of user attention hierarchy for personalized searches and recommendations.

REFERENCES

[1] K. Sugiyama, K. Hatano, and M. Yoshikawa, "Adaptive web search based on user profile constructed without any effort from users," in *Proceedings of the 13th conference on World Wide Web - WWW '04*, 2004, p. 675.

[2] Y. Wang and W. Shang, "Personalized news recommendation based on consumers' click behavior," in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2015*, 2016, pp. 634–638.

[3] R. Sieg, Ahu and Mobasher, Bamshad and Burke, "Improving the Effectiveness of Collaborative Recommendation with Ontology-Based User Profiles," in *proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, 2010, pp. 39–46.

[4] H. R. Kim and P. K. Chan, "Learning implicit user interest hierarchy for context in personalization," *Applied Intelligence*, vol. 28, no. 2, pp. 153–166, 2008.

[5] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure, "Ontological user profiling in recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 54–88, 2004.

[6] A. Sieg, B. Mobasher, and R. Burke, "Ontological User Profiles for Personalized Web Search," *Information Systems Journal*, vol. 105, no. Gruber 1993, pp. 84–91, 2007.

[7] V. Eyharabide and A. Amandi, "Ontology-based user profile learning," *Applied Intelligence*, vol. 36, no. 4, pp. 857–869, 2012.

[8] W. Shen, J. Wang, P. Luo, and M. Wang, "Linking named entities in Tweets with knowledge base via user interest modeling," *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13*, p. 68, 2013.

[9] L. Li, L. Zheng, F. Yang, and T. Li, "Modeling and broadening temporal user interest in personalized news recommendation," *Expert Systems with Applications*, vol. 41, no. 7, pp. 3168–3177, 2014.

[10] F. Liu, C. Yu, and W. Meng, "Personalized web search by mapping user queries to categories," in *Proceedings of the eleventh international conference on Information and knowledge management - CIKM '02*, 2002, p. 558.

[11] P. A. Chirita and C. Kohlsch, "Using ODP Metadata to Personalize Search Categories and Subject Descriptors," *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 178—-185, 2005.

[12] M. Michelson and S. a. Macskassy, "Discovering users' topics of interest on twitter: a first look," *AND '10: Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, pp. 73–80, 2010.

[13] P. Kapanipathi, P. Jain, C. Venkataramani, and A. Sheth, "User interests identification on Twitter using a hierarchical knowledge base," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8465 LNCS, 2014, pp. 99–113.

[14] K. W.-T. Leung, D. L. Lee, W. Ng, and H. Y. Fung, "A framework for personalizing web search with concept-based user profiles," *ACM Transactions on Internet Technology*, vol. 11, no. 4, pp. 1–29, 2012.

[15] Y. Xu, K. Wang, B. Zhang, Z. Chen, and K. Wang, "Privacy-enhancing personalized web search," *Proceedings of the 16th international conference on World Wide Web - WWW '07*, p. 591, 2007.

[16] H.-J. Kang, Dongyeop and Jiang, Daxin and Pei, Jian and Liao, Zhen and Sun, Xiaohui and Choi, "Multidimensional mining of large-scale search logs: a topic-concept cube approach," in *Proceedings of the fourth ACM international conference on Web search and data mining*, 2011, pp. 385—-394.

[17] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of community hierarchies in large networks," *Journal of Statistical Mechanics: Theory and Experiment (JSTAT)*, pp. 1–6, 2008.

[18] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 2 2, 2004.

[19] C. Lu, W. Lam, and Y. Zhang, "Twitter User Modeling and Tweets Recommendation Based on Wikipedia Concept Graph," *Workshops at the Twenty-Sixth AAAI Conference*, pp. 33–38, 2012.

Publication 2

Learning and inference in knowledge-based probabilistic model for medical diagnosis

Jingchi Jiang, Xueli Li, Chao Zhao, Yi Guan, and Qiubin Yu

Knowledge-Based Systems 138 (2017) 58–68

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Learning and inference in knowledge-based probabilistic model for medical diagnosis

Jingchi Jiang[a], Xueli Li[b], Chao Zhao[a], Yi Guan[a,*], Qiubin Yu[c]

[a] School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China
[b] EBAONET Healthcare Information Technology (Beijing) CO.LTD, Beijing 100028, China
[c] Medical Record Room, Second Affiliated Hospital of Harbin Medical University, Harbin 150086, China

ARTICLE INFO

Article history:
Received 19 September 2016
Revised 21 September 2017
Accepted 24 September 2017
Available online 25 September 2017

Keywords:
Probabilistic model
First-order knowledge
Markov network
Gradient descent
Markov logic network

ABSTRACT

Based on a weighted knowledge graph to represent first-order knowledge and combining it with a probabilistic model, we propose a methodology for creating a medical knowledge network (MKN) in medical diagnosis. When a set of evidence is activated for a specific patient, we can generate a ground medical knowledge network that is composed of evidence nodes and potential disease nodes. By incorporating a Boltzmann machine into the potential function of a Markov network, we investigated the joint probability distribution of the MKN. To consider numerical evidence, a multivariate inference model is presented that uses conditional probability. In addition, the weights for the knowledge graph are efficiently learned from manually annotated Chinese Electronic Medical Records (CEMRs) and Blood Examination Records (BERs). In our experiments, we found numerically that an improved expression of evidence variables is necessary for medical diagnosis. Our experimental results comparing a Markov logic network and six kinds of classic machine learning algorithms on the actual CEMR database and BER database indicate that our method holds promise and that MKN can facilitate studies of intelligent diagnosis.

1. Introduction

The World Health Organization (WHO) reports that 422 million adults have diabetes, and 1.5 million deaths are directly attributed to diabetes each year [1]. Additionally, the number of deaths caused by cardiovascular diseases (CVDs) and cancer annually is estimated to be 17.5 million and 8.2 million, respectively [2]. The WHO report on cancer shows that the number of new cases of cancer will increase by 70% over the next two decades. In the face of this situation, researchers have begun to pay more attention to health care. According to existing studies, more than 30% of cancer deaths could be prevented by early diagnosis and appropriate treatment [3]. Because an accurate diagnosis contributes to a proper choice of treatment and subsequent cure, medical diagnosis plays a significant role in improving health care. Consequently, a means to provide an effective intelligent diagnostic method to assist clinicians by reducing the costs and improving the accuracy of diagnosis has been a critical goal in the efforts to enhance the patient medical service environment.

Classification is one of the most widely researched topics in medical diagnosis. The general model classifies a set of symptom data into one of several predefined categories of disease for cases of medical diagnosis. A decision tree [4,5] is a classic algorithm in the medical classification domain, one that uses the information entropy method; however, it is sensitive to inconsistencies in the data. The support vector machine [6–8] has a solid theoretical basis for the classification task; because of its efficient selection of features, it has higher predictive accuracy than decision trees. Bayesian networks [9,10], which are based on Bayesian theory [11,12], describe the dependence relationship between the symptom variables and the disease variables; these can be used in medical diagnosis. Other diagnostic models include neural networks (NN) [13–15], fuzzy logic (FL) [16,17], and genetic algorithms (GAs) [18–20]. Each of these is designed with a distinct methodology for addressing diagnosis problems.

The existing studies have mainly focused on exploring effective methods for improving the accuracy of disease classification. However, these methods often ignore the importance of the application of domain knowledge. Although a hybrid Markov logic network (HMLN) [21], which is a generalization of a Markov logic network (MLN) [22], aims to integrate Boolean and numerical variables into a probabilistic logic modeling framework, inference is typically done by estimation methods that are based on variable

* Correspondence address.
E-mail addresses: jiangjingchi0118@163.com (J. Jiang), xueli.li@ebaonet.cn (X. Li), guanyi@hit.edu.cn (Y. Guan), yuqiubin6695@163.com (Q. Yu).

https://doi.org/10.1016/j.knosys.2017.09.030
0950-7051/

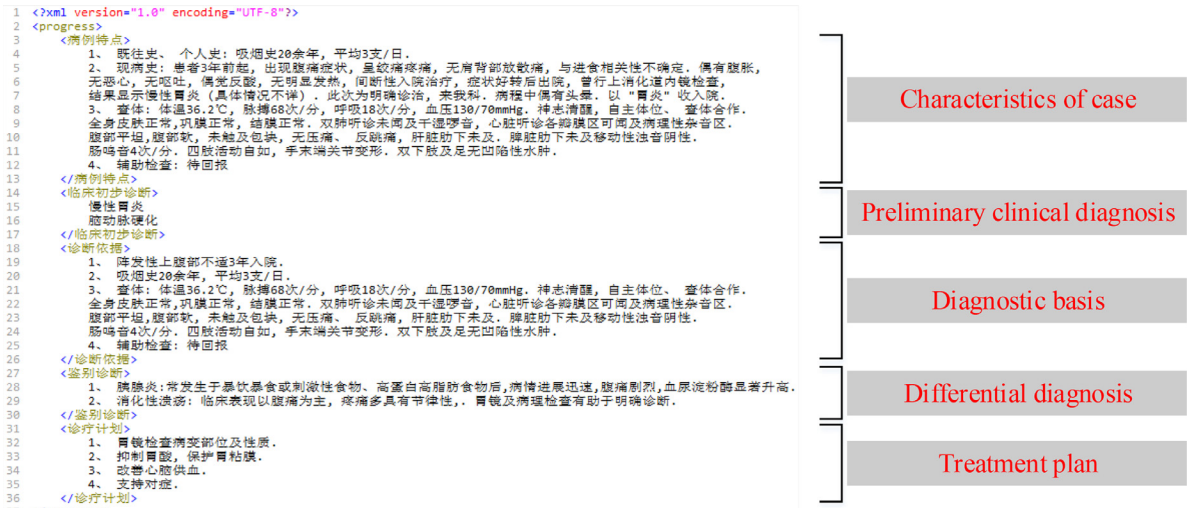


Fig. 1. Sample of progress note from the Second Affiliated Hospital of Harbin Medical University.

approximations or sampling strategies [23,24]. In addition, the inference efficiency of MLN and HMLN decreases steadily as the number of constraints increases. This is a significant problem because inference can become intractable for certain types of domain knowledge [25,26], especially in the health care domain, which contains a large amount of medical knowledge. In this paper, we focus on combining medical knowledge with a novel probabilistic model to assist clinicians in making intelligent decisions and show how this novel probabilistic model can be applied in medical diagnosis. We conducted our investigation as follows:

(1) Based on Chinese Electronic Medical Records (CEMRs) and Blood Examination Records (BERs), we developed techniques for the recognition of named entities and entity relationships. According to the relational structure between named entities, we built a symptom–disease knowledge base and an examination–disease knowledge base, each consisting of a set of rules in first-order logic.
(2) We mapped each first-order knowledge base into a knowledge graph. Each graph is composed of first-order predications (nodes) and diagnostic relationships among predications (edges). Furthermore, the graph can also be an intuitive reflection of the inferential structure of the knowledge.
(3) We developed a novel probabilistic model for medical diagnosis that is based on Markov network theory. To adapt it to the requirements of the multivariate feature, we incorporated a Boltzmann machine into the potential function of the Markov network. It can simultaneously model both binary and numerical variables. The mathematical derivation of learning and inference is rigorously deduced.
(4) By a numerical comparison with other diagnostic models for CEMRs and BERs, we found that our probabilistic model is more effective for diagnosing several diseases according to the measure of precision for the first 10 results (P@10) and that of recall for the first 10 results (R@10).

The rest of this paper is organized as follows. In Section 2, we introduce Chinese Electronic Medical Records, Blood Examination Records, and the knowledge graph. In Section 3, we review the fundamentals of Markov networks and Markov logic networks. In Section 4, the knowledge-based probabilistic model based on Markov networks is proposed; then, we demonstrate the mathematical derivation of learning and inference. In Section 5, we further evaluate the effectiveness and accuracy of our probabilistic model for medical diagnosis. Finally, we conclude this paper and discuss directions for future work in Section 6.

2. Knowledge extraction and knowledge representation

2.1. Chinese electronic medical records and blood examination records

Electronic medical records (EMRs) [27] are a systematized collection of patient health information in a digital format. As the crucial carrier of recorded medical activity, EMRs contain significant medical knowledge [28,29]. Therefore, for this study, we adopted Chinese Electronic Medical Records (CEMRs) in free-form text as the primary source of medical knowledge. These CEMRs, which have had protected health information (PHI) [30] removed, come from the Second Affiliated Hospital of Harbin Medical University, and we obtained the usage rights for research. These CEMRs include five main kinds of free-form text: discharge summaries, progress notes, patient complaints, patient disease histories, and communication logs. Considering the abundance of medical knowledge and the difficulty of Chinese text processing, we chose the discharge summaries and the progress notes as the sources for knowledge extraction. The structures of the progress note and discharge summary are shown in Figs. 1 and 2, respectively.

In contrast to the structure of CEMRs, BERs describe a series of blood test results, such as the levels of hemoglobin, alanine aminotransferase, and hepatitis B virus surface antigen. Most of the blood examination items use numerical evaluation criteria to make up for the shortage of discrete symptom variables in CEMRs. Furthermore, symptom-based preliminary diagnosis and examination-based definitive diagnosis are two essential components of the medical process. Therefore, BERs are important in disease diagnosis because they enable clinicians to make an accurate diagnosis based on the deviation of examination results from their respective normal ranges. In this study, BERs came from XingYi People's Hospital, and we obtained the usage rights for research. The structure of a BER, with one line per examination item, is shown in Fig. 3.

2.2. Corpus

The recognition of named entities [31] and entity relationships [32] is an important aspect in the extraction of medical knowledge from CEMRs. Referencing the medical concept annotation guideline and the assertion annotation guideline given by Informatics for Integrating Biology and the Bedside (i2b2) [33], we have drawn

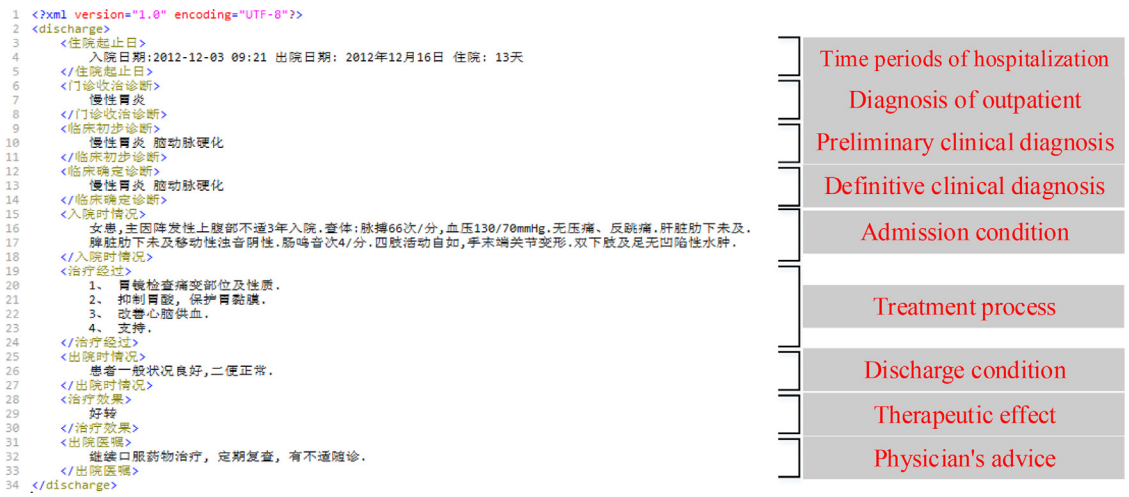


Fig. 2. Sample of discharge summary from the Second Affiliated Hospital of Harbin Medical University.


```
"TESTDATE":"20120214","ITEMCODE":"Glu","ITEMNAME":"血糖","RANGE":"3.89~~6.11","RESULT":"4.89","PATIENTID":"0000053104","FLAG":"正常","RANGEMEMO":"3.89~~6.11"}
"TESTDATE":"20120214","ITEMCODE":"Mg","ITEMNAME":"镁","RANGE":"0.73~~1.2","RESULT":"1.25","PATIENTID":"0000053104","FLAG":"高","RANGEMEMO":"0.73~~1.20"}
"TESTDATE":"20120214","ITEMCODE":"PHOS","ITEMNAME":"无机磷","RANGE":"0.81~~2.26","RESULT":"1.22","PATIENTID":"0000053104","FLAG":"正常","RANGEMEMO":"0.81~~2.26"}
"TESTDATE":"20120214","ITEMCODE":"CO2CP","ITEMNAME":"二氧化碳","RANGE":"21~~29","RESULT":"24.4","PATIENTID":"0000053104","FLAG":"正常","RANGEMEMO":"21~~29"}
"TESTDATE":"20120214","ITEMCODE":"Na","ITEMNAME":"钠","RANGE":"135~~147","RESULT":"141.4","PATIENTID":"0000053104","FLAG":"正常","RANGEMEMO":"135~~147"}
"TESTDATE":"20120214","ITEMCODE":"K","ITEMNAME":"钾","RANGE":"3.5~~5.5","RESULT":"3.87","PATIENTID":"0000053104","FLAG":"正常","RANGEMEMO":"3.5~~5.5"}
"TESTDATE":"20120214","ITEMCODE":"Cl","ITEMNAME":"氯","RANGE":"96~~108","RESULT":"106.9","PATIENTID":"0000053104","FLAG":"正常","RANGEMEMO":"96~~108"}
"TESTDATE":"20120215","ITEMCODE":"ALT","ITEMNAME":"谷丙转氨酶","RANGE":"0~~40","RESULT":"28","PATIENTID":"0000053614","FLAG":"正常","RANGEMEMO":"0~~40"}
"TESTDATE":"20120215","ITEMCODE":"AST","ITEMNAME":"谷草转氨酶","RANGE":"0~~37","RESULT":"23","PATIENTID":"0000053614","FLAG":"正常","RANGEMEMO":"0~~37"}
"TESTDATE":"20120215","ITEMCODE":"GGT","ITEMNAME":"γ-谷氨酰转移酶","RANGE":"0~~50","RESULT":"35","PATIENTID":"0000053614","FLAG":"正常","RANGEMEMO":"0~~50"}
"TESTDATE":"20120215","ITEMCODE":"TBIL","ITEMNAME":"总胆红素","RANGE":"5.1~~28","RESULT":"19.7","PATIENTID":"0000053614","FLAG":"正常","RANGEMEMO":"5.1~~28"}
"TESTDATE":"20120215","ITEMCODE":"DBIL","ITEMNAME":"直接胆红素","RANGE":"0~~10","RESULT":"3.8","PATIENTID":"0000053614","FLAG":"正常","RANGEMEMO":"0~~10"}
"TESTDATE":"20120215","ITEMCODE":"CHE","ITEMNAME":"胆碱脂酶","RANGE":"203~~460","RESULT":"631","PATIENTID":"0000053614","FLAG":"高","RANGEMEMO":"203~~460"}
"TESTDATE":"20120215","ITEMCODE":"UREA","ITEMNAME":"尿素","RANGE":"1.7~~8.3","RESULT":"4.42","PATIENTID":"0000053614","FLAG":"正常","RANGEMEMO":"1.7~~8.3"}
"TESTDATE":"20120215","ITEMCODE":"CREA","ITEMNAME":"肌酐","RANGE":"53~~123","RESULT":"82.9","PATIENTID":"0000053614","FLAG":"正常","RANGEMEMO":"44~~123"}
```


Fig. 3. Sample of BERs from XingYi People's Hospital.

on the guidelines for CEMRs [34,35] and manually annotated the named entity and entity relationships of 992 CEMRs as the resource of medical knowledge. For this diagnostic task, this study only kept "symptom" entities, "disease" entities, and the "indication" relationship. The "indication" relationship holds when the related "symptom" indicates that the patient suffers from the related "disease." In addition, there are seven modifiers for "symptom" entities, namely *present*, *occasional*, *conditional*, *historical*, *possible*, *absent*, and *not associated with the patient*.

2.3. Knowledge graph

The symptom–disease medical knowledge obtained from the 992 CEMRs can be comprehended as a set of first-order logic rules among "symptom" and "disease" entities. The reliability of the medical knowledge corresponds to the probability of the "indication" relationship. By gathering all the annotated "indication" relationships, a medical knowledge base may be constructed. However, the medical knowledge base lacks the connectivity of real-world knowledge. To capture this medical knowledge more intuitively, we build a more comprehensive knowledge graph that consists of "symptom" and "disease" entities as nodes and the "indication" relationships as edges. As the reliability of medical knowledge increases, the corresponding edge's weight grows gradually. The topology of the symptom–disease knowledge graph is shown in Fig. 4.

The nodes in the knowledge graph are divided into two colors according to the type of entity, with the red and green nodes representing "symptom" and "disease" entities respectively. This graph contains 173 kinds of diseases and 508 kinds of symptoms. As a whole, 1069 pieces of knowledge are represented in the symptom–disease knowledge graph.

In actual medical diagnosis, the answers that patients provide to questions about their symptoms can only be regarded as a preliminary clinical diagnosis because the complaint symptoms expressed by patients are not precise or scientific. In this case, clinicians typically order supplementary examinations to verify the preliminary diagnosis. Thus, examination–disease knowledge is important for medical diagnosis. In this study, the examination–disease knowledge is abstracted from BERs, each of whose examination items was recommended by clinicians. It incorporates rich clinical experience and has a noticeable correlation with the confirmed disease. Therefore, a second kind of knowledge graph, consisting of the recommended examination items and the confirmed diseases, can be established on the basis of "confirmation" relation-

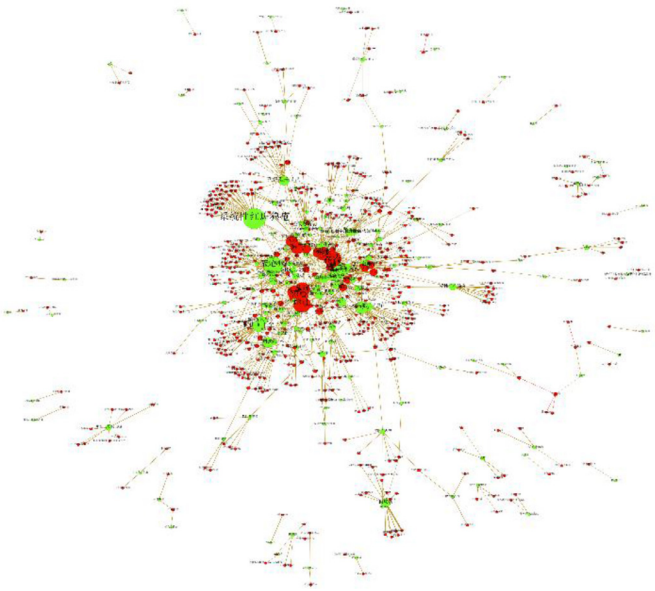


Fig. 4. Topology of the symptom–disease knowledge graph.

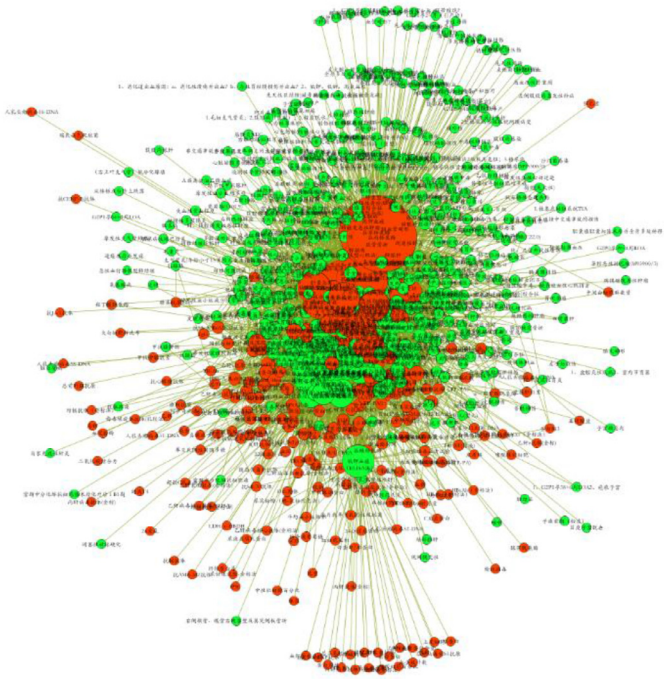


Fig. 5. Topology of the examination–disease knowledge graph.

ships. There are 494 kinds of disease and 220 kinds of examination in this graph. The topology of the examination–disease knowledge graph is shown in Fig. 5.

3. Markov logic networks (MLNs)

As a uniform framework of statistical relational learning, a MLN combines first-order logic with a probability graph model for solving problems of complexity and uncertainty. From a probability-and-statistics point of view, MLN is based on the methodology of Markov networks (MNs) [36]. From a first-order-logic point of view, it can briefly present the uncertainty rules and can tolerate incomplete and contradictory problems in the knowledge areas.

3.1. Markov networks and first-order logic

A Markov network, which is a model for the joint distribution of a set of variables $X = (X_1, X_2, ..., X_n) \in \chi$, provides the theoretical basis for a Markov logic network. A Markov network is composed of an undirected graph G and a set of potential functions ϕ_k. The joint distribution of the Markov network is given as

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(x_{\{k\}}) \tag{1}$$

where $x_{\{k\}}$ is the state of the kth clique. Z, known as the normalization function, is given by $Z = \sum_{x \in \chi} \prod_k \phi_k(x_{\{k\}})$. The most widely used method for approximate inference in MN is Markov chain Monte Carlo (MCMC), and Gibbs sampling in particular. Another popular inference method in MN is the sum-product algorithm.

A medical knowledge base is a set of rules in first-order logic. Rules are composed using four types of symbols: constants, variables, functions, and predicates. A term is any expression representing an object. An atom is a predicate symbol applied to a tuple of terms. A ground atom is an atomic rule, all of whose arguments are ground terms. A possible world assigns a truth value to each possible ground atom.

3.2. Markov logic networks in medical diagnosis

A Markov logic network can be considered as a template for generating Markov networks. Given different sets of constants, it will generate different Markov networks. According to the definition of a Markov network, the joint distribution of a Markov logic network is given by

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_i \omega_i n_i(x)\right) = \frac{1}{Z} \prod_i \phi_i(x_{\{i\}})^{n_i(x)} \tag{2}$$

where $n_i(x)$ is the number of true groundings of the ith rule R_i in constants x; ω_i is the weight for R_i; $x_{\{i\}}$ is the state of the atoms appearing in R_i; and $\phi_i(x_{\{i\}}) = e^{\omega_i}$. Because MLN only focuses on binary features, the constants x are discrete values and $x \in \{0, 1\}$.

To apply MLN in medical diagnosis, the atom is considered as the medical entity. When the medical entity presents an explicit condition for a patient, the corresponding atom of this entity in MLN is assigned the value 1; otherwise it is 0. By this mapping method, we are able to convert medical knowledge into the binary rules of the MLN. As medical knowledge accumulates, a MLN will be built, and its maximum probability model can be used for medical diagnosis. Given a series of symptoms, the risk probability for a specific disease is calculated by

$$\arg\max_y P(y|x) = \arg\max_y \sum_i \omega_i n_i(x, y) \tag{3}$$

Because of the higher complexity of calculating $n_i(x, y)$, the problem of maximum probability can be transformed into a satisfiability problem, for which a set of variables is searched to maximize the number of rules satisfied.

4. Methodology

Although MLN can be used for medical diagnosis, it is only suitable for binary rules because the values of $n_i(x, y)$ are uncountable when x is a continuous variable. Even though a hybrid Markov logic network (HMLN) can address numerical variables, the sum $S(x)$ of the values of all groundings also faces the computational intractability problem. No exact inference methods have been developed for HMLNs, primarily because of their excessive complexity. In particular, in the health care field, general inference methods require a set of quite complex and rich medical knowledge to be applied. When thousands of pieces of medical knowledge are generated, it is difficult to complete a disease diagnosis based on HMLN within a reasonable amount of time. Thus, the existing MLN methodology and its extensions have some obvious shortcomings for numeric-based diagnosis and are inefficient when there is a large number of multivariate rules. By changing the form of expression of the potential function, we can incorporate the continuous variable x into the joint distribution of MN, thereby enabling the conditional probability model for inference to be deduced via a Boltzmann machine, and the learning model for calculating the weight for each rule is proposed.

4.1. Medical knowledge network (MKN)

Based on the previously mentioned knowledge graph, we propose a model for handling numeric-based diagnosis that combines the knowledge graph with the theoretical basis of Markov networks. The novel theoretical framework is named the medical knowledge network (MKN).

Definition 1. A medical knowledge network L is a set of pairs (R_i, ω_i), where R_i is the medical knowledge in first-order logic and ω_i is the reliability of R_i. Together with a finite set of constants

$C = \{c_1, c_2, ..., c_n\}$, it defines a ground medical knowledge network $M_{L,C}$ as follows:

1. $M_{L,C}$ contains one multivariate node for each possible grounding of each medical entity appearing in L. The value of the node is the quantified indicator of the symptom or examination.
2. $M_{L,C}$ contains one weight for each piece of medical knowledge. This weight is the ω_i associated with R_i in L.

In a Markov network, a potential function is a nonnegative real-valued function of the state of the corresponding clique. Therefore, the potential function of MKN can also be regarded as the state of a clique, which is composed of the "indication" and "confirmation" relationships. Incorporating the quantified indicator of each node into the potential function is an important step for numeric-based diagnosis. From statistical physics, we can express the potential function as an energy function [37], rewriting $\phi(D)$ as

$$\phi(D) = \exp(-\varepsilon(D)) \tag{4}$$

where $\varepsilon(D)$ is often called an energy function. The set D is the state of the "symptom" and "disease" nodes, or the state of the "examination" and "disease" nodes. Then, the expression $\varepsilon(D)$ is interpreted in terms of an unrestricted Boltzmann machine [38], which is the one of the earliest types of Markov network. The energy function associated with the "indication" and "confirmation" relationships is defined by a particularly simple parametric form:

$$\varepsilon(D) = \varepsilon_{ij}(x_i, x_j) = -\omega_{ij} x_i x_j \tag{5}$$

where x_i and x_j represent the values of the respective nodes of set D, and ω_{ij} is the contribution of the energy function. According to Eq. (1), the joint distribution is defined as follows:

$$\begin{aligned} P(X = x) &= \frac{1}{Z} \prod_{\substack{i \\ r_i \in R}} \phi_i(D) = \frac{1}{Z} \prod_{\substack{i \\ r_i \in R}} \exp(-\varepsilon(D)) \\ &= \frac{1}{Z} \exp\left(- \sum_{\substack{i \\ r_i \in R}} (-\omega_i x^s_{r_i} x^d_{r_i}) \right) \\ &= \frac{1}{Z} \exp\left(\sum_{\substack{i \\ r_i \in R}} \omega_i x^s_{r_i} x^d_{r_i} \right) \end{aligned} \tag{6}$$

In general, the energy function of an unrestricted Boltzmann machine contains a set of parameters u_i that encode individual node potentials. These activated individual variables will stress the effects of the "symptom" and "examination" in the energy function. The rewritten probability formula is given as

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_{\substack{i \\ r_i \in R}} (\omega_i x^s_{r_i} x^d_{r_i} + u_{x^s_{r_i}} x^s_{r_i}) \right) \tag{7}$$

As seen, when a "symptom" or "examination" is activated, the factor of the corresponding individual node potential will be considered a major component of the model; this is exactly consistent with a clinical diagnosis that is based on symptoms. In this paper, we adopt the Gaussian potential function (GPF) as the individual node potential, which is expressed as

$$u_{x_{r_i}} = \sum_{j=1}^{n} \left(m_{x^d_j} e^{-\left(\frac{d_{x^s_{r_i} x^d_j}}{\sigma} \right)^2} \right) \tag{8}$$

where $d_{x^s_{r_i} x^d_j}$ represents the distance between node $x^s_{r_i}$ and its neighboring node x^d_j in the knowledge graph. The influence factor σ is used to control the influence range of each node, and $m_{x^d_j}$ is the quality of node x^d_j. The final probability model is defined as the following distribution:

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_{\substack{i \\ r_i \in R}} \left(\omega_i x^s_{r_i} x^d_{r_i} + \left(\sum_{j=1}^{n} m_{x^d_j} e^{-\left(\frac{d_{x^s_{r_i} x^d_j}}{\sigma} \right)^2} \right) \cdot x^s_{r_i} \right) \right) \tag{9}$$

Algorithm 1
Construction of symptom–disease medical knowledge network.

Input: $List_{EMR}$: a list of the electronic medical records for training.
Output: *Network*: a medical knowledge network.
Begin
1: Initialize the lists of nodes Set_{node} and list of edges Set_{edge} in MKN.
2: Extract the entities and relationships from the $List_{EMR} \rightarrow Rules = \{rule_1, rule_2, ..., rule_n\}$.
3: Initialize the weights for *Rules* by a fixed value ω.
4: ***for*** $rule_i \in Rules$ ***do***
5: Initialize the symptom node $Node_{symptom}$ and disease node $Node_{disease}$.
6: Parse the symptom predicate and the disease predicate from $rule_i \rightarrow Node_{symptom}$ and $Node_{disease}$.
7: $Set_{node} \leftarrow Node_{symptom}$.
8: $Set_{node} \leftarrow Node_{disease}$.
9: Define the relationship $Edge_i$ between $Node_{symptom}$ and $Node_{disease}$.
10: Add $Edge_i$ to Set_{edge}, and assign ω as the weight of $Edge_i$.
11: ***end for***
12: ***Function*** *PageRank*(Set_{node}, Set_{edge})
13: After calculating the PageRank of all nodes, the MKN is built: Set_{node}, $Set_{edge} \rightarrow Network$.
14: ***return*** *Network*.

Using Definition 1 and the deduced joint distribution, Algorithm 1 provides the procedure for building a medical knowledge network.

By traversing the rules and parsing the predicates, a medical knowledge network can be implemented. The PageRank function is used as an indicator of the quality of each node. To characterize the reliability of medical knowledge, we set a fixed value ω for the initial network.

4.2. Inference

Medical inference can answer two common generic clinical questions: "What is the probability that rule R_1 holds given rule R_2?" and "What is the probability of disease D_1 given the symptom vector S_1 or examination vector E_1?" In response to the first problem, we can answer by computing the conditional probability as

$$\begin{aligned} P(R_1 | R_2, L, C) &= P(R_1 | R_2, M_{L,C}) \\ &= \frac{P(R_1 \wedge R_2 | M_{L,C})}{P(R_2 | M_{L,C})} \\ &= \frac{\sum_{x \in \chi_{R_1} \cap \chi_{R_2}} P(X = x | M_{L,C})}{\sum_{x \in \chi_{R_2}} P(X = x | M_{L,C})} \end{aligned} \tag{10}$$

The set χ_{R_i} is the set of rules where R_i holds, and $P(x|M_{L,C})$ is given by Eq. (9). Through the free combinations of pairs of disconnected atoms in MKN, some new rules will be derived by inference. When the probability of a new rule exceeds a certain threshold, it can be concluded that the new rule is reliable under the current knowledge base. Rule inference not only helps enrich the knowledge base but is also a self-learning mechanism for the MKN.

The second inference question is what is usually meant by "disease diagnosis." On the condition that the patient has a given symptom vector or examination vector, we can predict the risk probability for a specific disease. This can be classified as a typical problem of conditional probability. The risk probability of disease y can be calculated by

$$P(Y=y|B_l=b_l) = \frac{\exp\left(\sum_{\substack{i \\ r_i\in R_l}}\left(\omega_i x_{r_i} y_{r_i} + \left(\sum_{j=1}^{n} m_{y_j} e^{-\left(\frac{d_{x_{r_i}y_j}}{\sigma}\right)}\right)\cdot x_{r_i}\right)\right)}{\exp\left(\sum_{\substack{i \\ r_i\in R_l}}\left(\omega_i x_{r_i} y^0_{r_i} + \left(\sum_{j=1}^{n} m_{y_j} e^{-\left(\frac{d_{x_{r_i}y_j}}{\sigma}\right)}\right)\cdot x_{r_i}\right)\right) + \exp\left(\sum_{\substack{i \\ r_i\in R_l}}\left(\omega_i x_{r_i} y^1_{r_i} + \left(\sum_{j=1}^{n} m_{y_j} e^{-\left(\frac{d_{x_{r_i}y_j}}{\sigma}\right)}\right)\cdot x_{r_i}\right)\right)} \tag{11}$$

where R_l is the set of ground rules in which disease y appears, and b_l is the Markov blanket of y. The Markov blanket of a node is the minimal set of nodes that renders it independent of the remaining network; this is simply the set of that node's neighbors in the knowledge graph. Corresponding to the ith ground rule, y_{r_i} is the value (0 or 1) of disease y. In contrast to the MLN diagnostic model, MKN avoids the complexity problem of $n_i(x, y)$ and incorporates the quantitative value x_{r_i} of symptom or examination into the diagnostic model. The detailed diagnostic algorithm is shown in Algorithm 2.

Following the Eq. (11), we propose a disease diagnostic algorithm based on MKN. To provide a reliable diagnosis, we need to calculate the risk of each disease. According to the evidences, Algorithm 2 can generate a list of potential disease that is sorted by diagnostic possibility.

4.3. Learning

A learning model is proposed to calculate the weight of each piece of medical knowledge. In this study, we adopted the gradient descent method. Assuming independence between diseases, the learning model first calculates the joint probability distribution of a disease vector:

$$P^*_\omega(Y=y) = \prod_{l=1}^{m} P_\omega(Y_l=y_l|M_{L,C}) \tag{12}$$

where m is the dimension of disease vector y. According to Eq. (12), the derivative of the log-likelihood function with respect to the weight for the ith rule is

$$\frac{\partial}{\partial\omega_i}\log P^*_\omega(Y=y) = \frac{\partial}{\partial\omega_i}\log\prod_{l=1}^{m} P_\omega(Y_l=y_l|M_{L,C}) = \sum_{l=1}^{m}\frac{\partial}{\partial\omega_i}\log P_\omega(Y_l=y_l|M_{L,C}) \tag{13}$$

Algorithm 2
Disease diagnostic algorithm based on MKN.

Inputs: *Rules*: a set of rules with the learned weights ω.
PR: a set of PageRank values for the nodes in MKN.
Evidences: a set of ground atoms with known values for a specific patient.
Query: a set of ground atoms with unknown disease values.
Output: *Result*: a diagnosis result for the specific patient.
Begin
for $disease_i \in Query$ ***do***
  Initialize the probability $Pro_{activated}$ with the activated $disease_i$.
  Initialize the probability $Pro_{inactivated}$ with the inactivated $disease_i$.
  //Activating the disease atoms in *Network*.
  for $rule_j \in Rules$ ***do***
    $Pro_{inactivated} += \omega_j \cdot symptom_j \cdot disease_j + PR_{disease_j} \cdot symptom_j/E$.
    if $disease_i \in rule_j$
      Activate the atom of $disease_j$.
    end if
  $Pro_{activated} += \omega_j \cdot symptom_j \cdot disease_j + PR_{disease_j} \cdot symptom_j/E$.
end for
$Result_i = exp(Pro_{activated})/(\exp(Pro_{activated}) + exp(Pro_{inactivated}))$.
end for
Function *Sort*(*Result*)
return *Result*.

The calculation of $\frac{\partial}{\partial\omega_i}\log P_\omega(Y_l=y_l|M_{L,C})$ will be a stubborn problem. Therefore, we try to construct the derivative of the log-likelihood. From Eq. (9), we know that the normalization function Z can be expressed as

$$Z = \sum\nolimits_{y\in\eta}\exp\left(\sum_{\substack{i \\ r_i\in R}}\left(\omega_i x_{r_i} y_{r_i} + \left(\sum_{j=1}^{n} m_{y_j} e^{-\left(\frac{d_{x_{r_i}y_j}}{\sigma}\right)}\right)\cdot x_{r_i}\right)\right) \tag{14}$$

Then, we have the pseudo-log-likelihood of Eq. (9) and its gradient:

$$\log P(Y{=}y|M_{L,C}) = \sum_{\substack{i \\ r_i\in R}}\left(\omega_i x_{r_i} y_{r_i} + \sum_{j=1}^{n}\left(m_{y_j} e^{-\left(\frac{d_{x_{r_i}y_j}}{\sigma}\right)}\right)\cdot x_{r_i}\right) - \log Z \tag{15}$$

$$\begin{aligned}\frac{\partial}{\partial\omega_i}\log P(Y=y|M_{L,C}) &= x_{r_i}y_{r_i} - \frac{1}{Z}\left(\sum\nolimits_{y\in\eta}\exp\left(\sum_{\substack{i \\ r_i\in R}}\left(\omega_i x_{r_i} y_{r_i} + \left(\sum_{j=1}^{n} m_{y_j} e^{-\left(\frac{d_{x_{r_i}y_j}}{\sigma}\right)}\right)\cdot x_{r_i}\right)\right)\cdot x_{ri} y_{r_i}\right) \\ &= x_{r_i}y_{r_i} - \sum\nolimits_{y\in\eta} P(Y=y|M_{L,C})\cdot x_{r_i}y_{r_i}\end{aligned} \tag{16}$$

where η is the set of all possible values of y, and $P(Y=y|M_{L,C})$ can be given by Eq. (9). By bringing Eq. (16) into Eq. (13), the derivative of the log-likelihood with respect to the weight for the ith rule can be naturally calculated. We get the final expression

$$\frac{\partial}{\partial\omega_i}\log P^*_\omega(Y=y) = \sum_{l=1}^{m}\left(x_{r_i}y_{r_i} - \sum\nolimits_{y\in\eta} P(Y=y|M_{L,C})\cdot x_{r_i}y_{r_i}\right) \tag{17}$$

After finite iterations, ω_i is calculated with the learning rate λ.

$$\omega_{i,t} = \omega_{i,t-1} + \lambda\frac{\partial}{\partial\omega_i}\log P^*_\omega(Y=y)|\omega_{t-1} \tag{18}$$

The detailed procedure for the weight learning model is presented in Algorithm 3.

In summary, we adopt the log-likelihood function and the gradient descent method to learn the weight vector. Fortunately, the gradient of pseudo-log-likelihood can be calculated by the joint probability distribution in finite time. Mapping the evidence to its Markov blanket is also to improve the time-effectiveness of the learning algorithm.

5. Experiments and discussion

To verify MKN's effectiveness, we conducted experiments using actual CEMRs and BERs. Based on the knowledge graph concept described in Section 2.3, we built two MKNs for medical diagnosis: the symptom–disease knowledge graph and the examination–disease knowledge graph.

Algorithm 3
Learning algorithm for MKN.

Inputs: *Network*: an MKN with vector ω of fixed weights.
Evidences: a set of ground atoms with known values.
Output: *Weights*: a learned weight vector.
Begin
Initialize the weight vector *Weights*.
for $weight_i \in Weights$ ***do*** //$weight_i$ represents the weight for the ith rule
  while t *From* 1 *To* 100 ***do***
    for $evidence_j \in Evidences$ ***do*** //evidence set for the jth patient
      Extract the blanket of the ith rule $\rightarrow blanket_i$.
      //Mapping $evidence_j$ to $blanket_i$
      $slope += s_{ij} \cdot d_{ij} - \sum_{x' \in X} [P_\omega(X_l = x' | blanket_i) \cdot s_{ij} \cdot d_{ij}]$.
      //s_{ij} represents the symptom value of $evidence_j$ for the ith rule
      //d_{ij} represents the disease value of $evidence_j$ for the ith rule
    end for
    $\omega_{i,t} = \omega_{i,t-1} + \eta \cdot slope$.
  end while
  $weight_i \leftarrow \omega_{i,t}$.
end for
return *Weights*.

Table 1
Assertions and their grades in MKN.

Grade	Assertion types
1	Absent & Not associated with the patient
2	Historical & Possible
3	Occasional & Conditional
4	Present

For the symptom–disease knowledge graph, we chose the manually annotated 992 CEMRs with the help of medical professionals and kept only the discharge summaries and progress notes as the sources of knowledge. In the annotation process, we classified the entities into five categories: disease, type of disease, symptom, test, and treatment; only the disease entity and the symptom entity were extracted to complete the diagnostic task. Additionally, owing to the lack of numerical indices for symptoms in our CEMRs, we adopted the modifiers *present*, *occasional*, *conditional*, *historical*, *possible*, *absent*, and *not associated with the patient*, to represent the symptom variable x, corresponding to four grades, as summarized in Table 1. Although the modifier of the symptom is not a continuous variable, a multivariate version of MKN also has theoretical significance.

After this MKN was constructed, we randomly selected 300 untagged CEMRs as the test corpus, and conditional random fields (CRFs) were used to automatically recognize the disease entities and symptom entities. Based on the symptom entities on each CEMR, we inferred the diagnosis result and ascertained whether there was consistency between the diagnosed disease and the actual disease.

For the examination–disease knowledge graph, we selected 6000 BERs as the source of examination–disease knowledge. Each BER includes the examination items, numerical results, corresponding normal ranges, and confirmed diseases of a specific patient. We extracted the examination items and confirmed the diseases to generate examination–disease knowledge, which is represented in triple form. Further, the numerical results and corresponding normal ranges are used as the input for a ground MKN. Finally, 4516 pieces of knowledge were extracted from 6000 BERs. Due to the explicitness of examination–disease relationships, which can be identified by regular expressions, the examination–disease knowledge graph was naturally created in an automated way, thereby avoiding the defects of manual annotation. In contrast to CEMRs, most examination items in BERs are represented in numerical form; furthermore, the standard range of each item is given. For follow-up experiments, 2000 BERs were randomly selected as the test corpus. From the examination results for each patient, we attempted to deduce diseases having the highest risk probability.

The description and analysis of the experiments are concerned with three aspects: the parameter analysis, the weight learning, and the effectiveness of MKN in the symptom–disease knowledge graph and the examination–disease knowledge graph.

5.1. Parameter analysis

In this section, we focus on the optimum choices for the parameter values. In Eq. (11), $d_{x_{r_i} y_j}$ is the distance between x_{r_i} and its neighboring node y_j in the knowledge graph. Therefore, we define $d_{x_{r_i} y_j} = 1$. The influence factor σ represents the control range of each node. If a symptom atom and a disease atom appear in a common rule, they have an interaction with each other, and the two atoms in each rule can be represented as two adjacent nodes in the knowledge graph. Since we naturally assume that the symptom node affects only the nearest connected disease node, we set $\sigma = 1$.

The terms m_{y_j}, which is the quality of the disease node, and x_{r_i}, which is the evidence variable in the ith rule, are both uncertainty parameters. The selection of expressions for m_{y_j} and x_{r_i} will affect the accuracy of MKN in diagnosing disease. To begin, we experimented with three classical measures for m_{y_j}: PageRank, degree, and betweenness centrality. We used the discounted cumulative gain (DCG) [39] as the indicator to measure the accuracy of the diagnosis result. The DCG score can be calculated by

$$DCG_P = rel_1 + \sum_{i=2}^{P} \frac{rel_i}{\log_2^i} \tag{19}$$

where rel_i represents the relevance of the ith disease in the diagnosis result; a correct diagnosis is 1, whereas a misdiagnosis is 0. The variable P is the number of diagnosis results, which in this study was 10.

As structural differences between the discharge summary and the progress note can lead to different numbers of symptoms for the same patient, two experiments were used to distinguish between them. Fig. 6 shows the DCG scores (y-axis) plotted against the serial numbers of 40 discharge summaries and 260 progress notes (respective x-axis) for the three measures of quality for the disease node.

Although the results show that the curves of the DCG scores are irregular, the DCG score is 1 in most cases. From the DCG descriptions, this occurs because most of the CEMRs have only one actual disease, so our model ranks this disease at the top of the diagnosis result. We also observe that the effectiveness of the PageRank-based MKN is better than those of the other methods for both the discharge summary and the progress note.

To describe the diagnosis result more directly, we adopt two other measures, R@10 and P@10, which are the recall and precision, respectively, for the first 10 results. If m actual diseases appear in a record and the MKN returns n of them, then the R@10 value is given by

$$R@10 = \frac{n}{m} \tag{20}$$

The mean recall is $\bar{R}@10 = \sum_l R@10/l$, where l denotes the number of test samples. In contrast to recall, precision is defined as the ratio of correctly diagnosed cases to the total number of cases:

$$\bar{P}@10 = \frac{r}{l} \tag{21}$$

Precision measures the diagnostic accuracy of MKN, and recall focuses on the diagnostic coverage. In addition, we use the

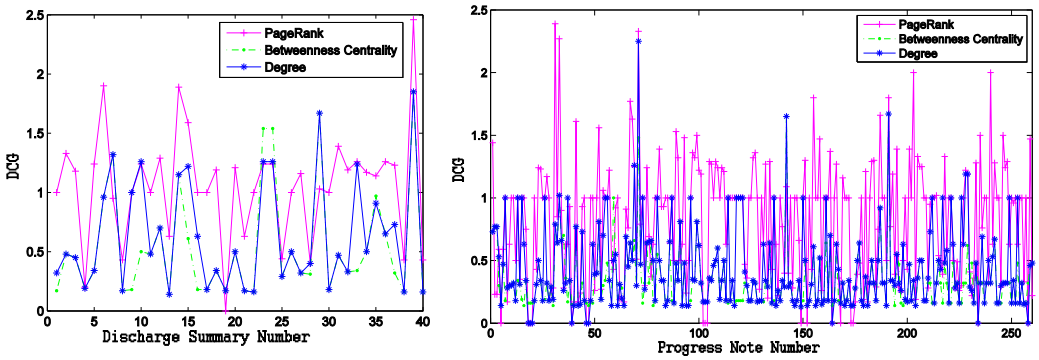


Fig. 6. DCG (discounted cumulative gain) for discharge summaries and progress notes using different measures of quality for the disease node.

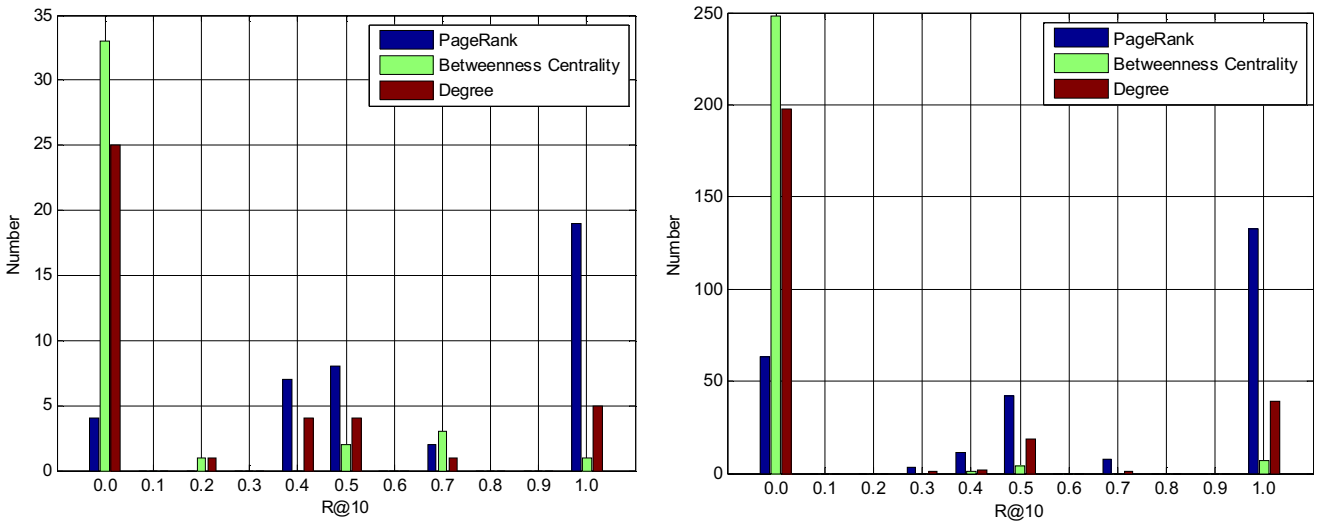


Fig. 7. Distribution of R@10 (recall for first 10 results) for discharge summaries (left) and progress notes (right) using different measures of quality for the disease node.

weighted harmonic mean of recall and precision, called the F-measure, as a comprehensive indicator to measure the effectiveness of MKN.

Fig. 7 shows the distributions of R@10 for discharge summaries and progress notes, with the blue, green, and red bars showing the results using PageRank, betweenness centrality, and degree, respectively. Under PageRank, the recall for nearly half the records is 1.0. By contrast, the results for betweenness centrality and degree are unsatisfactory because they have higher proportions with a recall of 0.0 and lower proportions with 1.0. Considering both factors DCG and R@10, we conclude that PageRank is more appropriate to use as the quality of disease node m_{y_j}.

The second uncertainty parameter is x_{r_i}, which is the quantitative value of the symptom or examination. For the diagnostic task, we designed an improved function to express x_{r_i}. We seek a representation of x_{r_i} that not only satisfies the requirements for discrete values but also might be suitable for continuous values. The improved quantitative function is defined as follows:

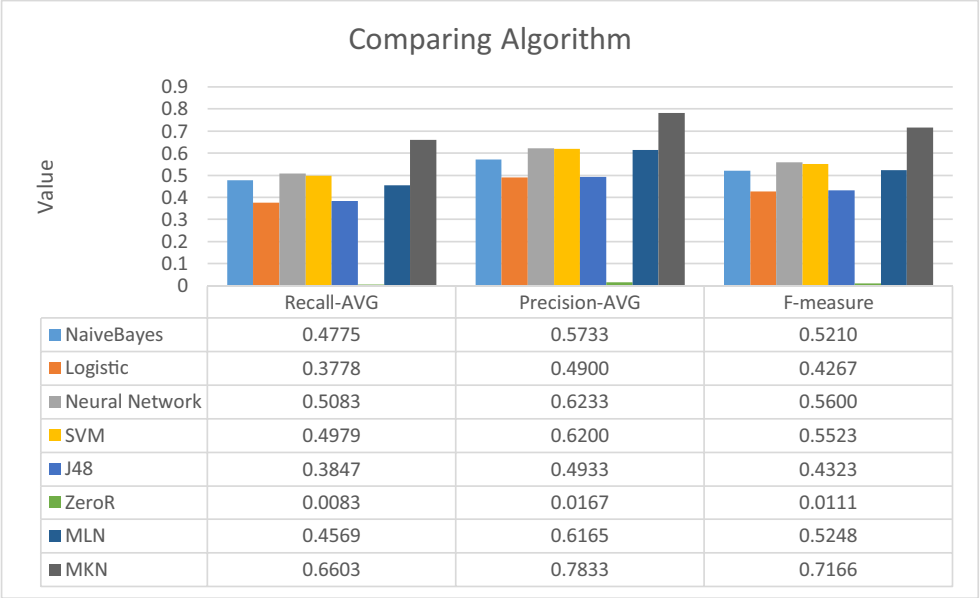
$$S(x) = \begin{cases} \left| \tan\left[\dfrac{\pi \cdot (x - x_{\min})}{4 \times (x_{normal-\min} - x_{\min})} - \dfrac{\pi}{4} \right] \right| & x < x_{normal-\min} \\ \left| \tan\left[\dfrac{\pi \cdot (x - x_{normal-\max})}{4 \times (x_{\max} - x_{normal-\max})} \right] \right| & x > x_{normal-\max} \end{cases} \tag{22}$$

where x is the value of the symptom or examination, and $x_{normal-\max}$ and $x_{normal-\min}$ are the upper and lower limits, respectively, of the normal range. With an increasing deviation between the variable x and the normal value, $S(x)$ grows exponentially, which also agrees with medical knowledge. More importantly, we can map the variable to a normalization interval, which is $0 < S(x) < 1$.

5.2. Weight learning

In this section, we make a credibility assumption: If a ground atom is in the knowledge base, it is assumed to be true; otherwise, it is false. In other words, the inference of the MKN depends completely on the existing medical knowledge. To test the effectiveness of the learning method, we compared three types of weighting: constant weighting, MLN-based weighting, and MKN-based weighting. We employed the learning program "Tuffy," [40] which is an open-source MLN inference engine. In addition, we experimented using different constants as weights to check whether it might influence the diagnosis results. Based on symptom-disease knowledge base, Tables 2 and 3 summarize the results for the discharge summaries and progress notes, respectively, showing P@10, precision for the first 20 results (P@20), R@10, and average DCG.

We can see that the constant weights of 0.5 and 1 give exactly the same results, demonstrating that the diagnosis results are not at all influenced by the weights' being equally adjusted. Then, we experimentally conclude that the weight learning methods in order of effectiveness are MKN-based, constant, positive MLN-based, and MLN-based.



	Recall-AVG	Precision-AVG	F-measure
NaiveBayes	0.4775	0.5733	0.5210
Logistic	0.3778	0.4900	0.4267
Neural Network	0.5083	0.6233	0.5600
SVM	0.4979	0.6200	0.5523
J48	0.3847	0.4933	0.4323
ZeroR	0.0083	0.0167	0.0111
MLN	0.4569	0.6165	0.5248
MKN	0.6603	0.7833	0.7166

Fig. 8. Comparison of eight diagnostic algorithms on 300 CEMRs (Chinese Electronic Medical Records): six machine learning methods, MLN (Markov logic network), and MKN (medical knowledge network).

Table 2
Analysis of effectiveness of weight learning for discharge summaries.

	Index			
Weight type	P@10	P@20	R@10	DCG-AVG
Constant weight of 0.5	0.875	0.9	0.62	1.06
Constant weight of 1	0.875	0.9	0.62	1.06
MLN weight	0.8056	0.8611	0.5233	0.822
MKN weight	0.9	0.95	0.67	1.0983

P@10 = precision for first 10 results; P@20 = precision for first 20 results; R@10 = recall for first 10 results; DCG-AVG = average discounted cumulative gain.
MLN = Markov logic network; MKN = medical knowledge network.

Table 3
Analysis of effectiveness of weight learning for progress notes.

	Index			
Weight type	P@10	P@20	R@10	DCG-AVG
Constant weight of 0.5	0.7538	0.8115	0.5949	0.7909
Constant weight of 1	0.7538	0.8115	0.5949	0.7909
MLN weight	0.6473	0.7593	0.5006	0.6743
MKN weight	0.7653	0.8692	0.6588	0.8286

P@10 = precision for first 10 results; P@20 = precision for first 20 results; R@10 = recall for first 10 results; DCG-AVG = average discounted cumulative gain.
MLN = Markov logic network; MKN = medical knowledge network.

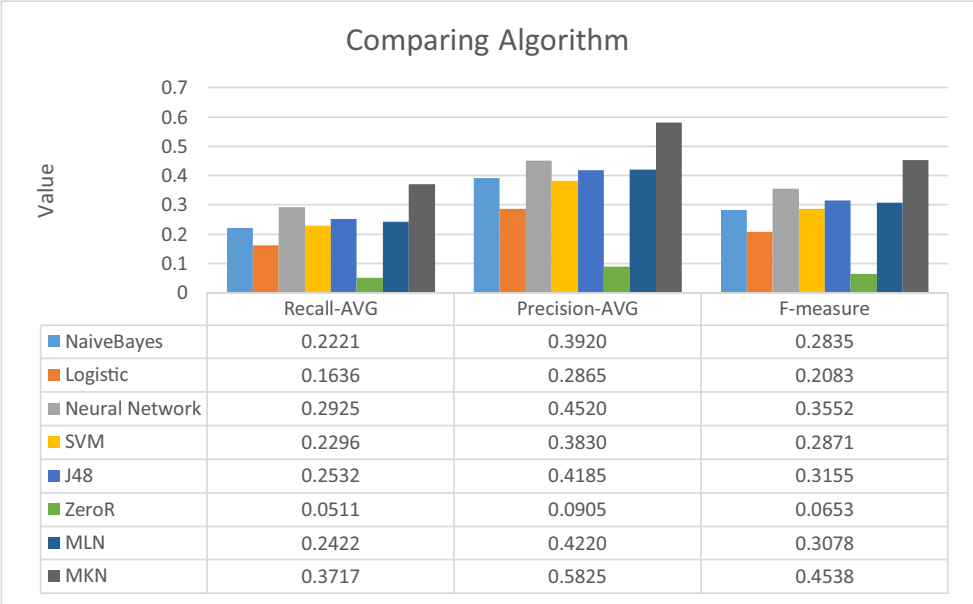
5.3. Comparison with other algorithms

After determining the uncertainty parameters and the type of weights, we compared MKN with existing diagnostic systems—MLN and six kinds of machine learning methods—using the symptom–disease knowledge base and the examination–disease knowledge base. Fig. 8 shows the performance of the machine learning methods and probabilistic graphical models with the symptom–disease knowledge base. MKN is clearly more accurate than the other methods are, thus demonstrating the promise of this approach. The traditional machine learning approach performs well on some CEMRs but very poorly on others; the recall values of these methods are uniformly poor. The experiment indicates that machine learning methods are difficult to apply for a diagnosis of multiple diseases, especially when a patient's symptoms are sparse. Compared with machine learning methods, there is a greater difference between MLN and MKN in terms of $\bar{P}$@10 and $\bar{R}$@10. We believe that the theoretical mechanism of MLN being based on rough binary logic is the main cause of the poor effect; the binary atoms cannot precisely capture the degree of seriousness of the symptoms. However, this experiment testifies to the advantage of MKN's multivariate atoms for medical diagnosis.

Overall, the best-performing diagnostic method is MKN, but approximately 20% of CEMRs are still completely misdiagnosed. The most likely reason is that the symptom–disease knowledge base created from 992 annotated CEMRs is minuscule. Thus, we also conducted examination-based diagnostic experiments using 8000 BERs.

In preparation for the examination-based comparison experiments, 6000 training BERs and 2000 testing BERs had to be preprocessed, which included extracting the scope of the standard for each examination item, merging the results of multiple examinations in multiple times for the same patient and similar operations. Similar to the presentation above of the results of symptom-based diagnosis, Fig. 9 compares the results of the examination-based diagnosis for eight diagnostic algorithms.

In Fig. 9, the same three indicators are used to evaluate the performance of the eight diagnostic algorithms. For the machine learning methods, the F-measure is generally low; most of the values are concentrated between 0.2 and 0.36. The experimental results indicate that the effectiveness of MKN is clearly superior to that of other algorithms. One of the key reasons for MKN's good performance is that MKN emphasizes the relationships between the "examination" and "disease" nodes. Consequently, the scope of potential disease can be significantly reduced by analyzing network connectivity when some "examination" nodes are activated. The top ten confirmed diseases can be ranked with a higher probability. Conversely, machine learning methods ignore the importance of domain knowledge. Their assumed independence between examination features is the primary cause of their misdiagnoses. In particular, in the field of medicine, the occurrence of a disease is the consequence of the combined contributions of several interrelated factors. Machine learning methods based on numerical features cannot reflect these complex relationships.



	Recall-AVG	Precision-AVG	F-measure
NaiveBayes	0.2221	0.3920	0.2835
Logistic	0.1636	0.2865	0.2083
Neural Network	0.2925	0.4520	0.3552
SVM	0.2296	0.3830	0.2871
J48	0.2532	0.4185	0.3155
ZeroR	0.0511	0.0905	0.0653
MLN	0.2422	0.4220	0.3078
MKN	0.3717	0.5825	0.4538

Fig. 9. Comparison of eight diagnostic algorithms on 2000 BERs (Blood Examination Records): six machine learning methods, MLN (Markov logic network), MKN (medical knowledge network).

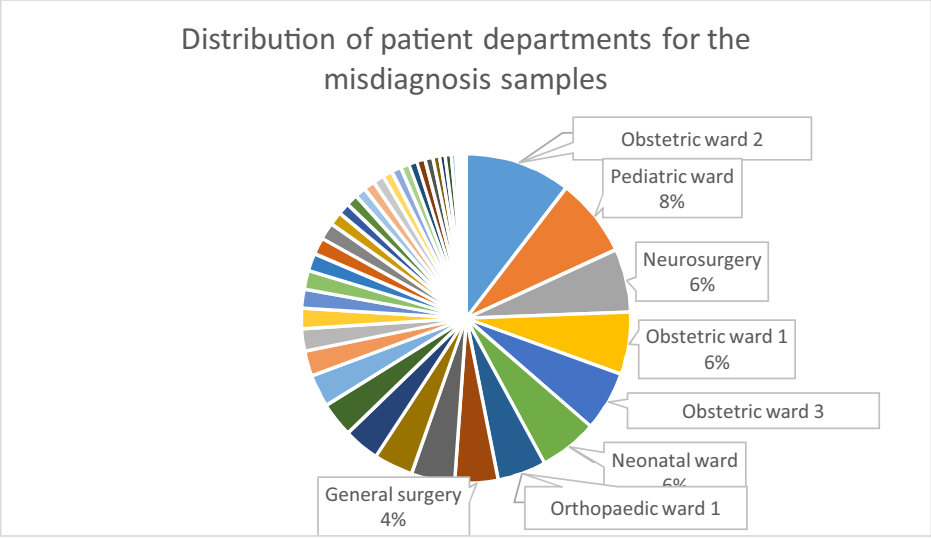


Fig. 10. Distribution of patient departments for misdiagnosed samples.

However, the effectiveness of our experimental results is far from what is expected. We further analyze the possible causes. One of the leading causes is that some diseases, such as surgical diseases and obstetrical diseases, cannot be properly diagnosed by using BERs as the only evidence because blood testing as a routine examination may not have any relationship with these diseases. Through the statistical distribution of patient departments for the misdiagnosis samples, this conjecture is verified. From Fig. 10, we can see that the top eight departments represented in misdiagnosis cases including the obstetrics department, the orthopedic department, and general surgery, are those that have only a weak relationship to blood testing.

6. Conclusion and future work

In this paper, we presented a knowledge-based probabilistic model for medical diagnosis. By extracting medical knowledge from Chinese Electronic Medical Records and Blood Examination Records, a symptom–disease knowledge graph composed of “disease” and “symptom” nodes and an examination–disease knowledge graph composed of “disease” and “examination” nodes were constructed. Building on the theory of Markov networks, we developed a novel probabilistic model, called the medical knowledge network (MKN).To address the problem of numeric-based diagnosis, the model applies the energy function of Boltzmann machines as the potential function. Then, the mathematical derivation processes of learning and inference were rigorously deduced. The medical knowledge network can adopt ternary rules or even continuous numerical rules. PageRank and our improved quantitative function were used to represent the symptom variables and the examination variables, respectively. Empirical tests with actual records illustrate that MKN can improve the diagnostic accuracy. Through comparisons with other algorithms, the effectiveness and promise of MKN were also demonstrated.

MKN is a knowledge-based inference model that is applicable to many AI problems but leaves ample space for future development. Directions for future work fall into three main areas:

6.1 Knowledge base

We plan to annotate more records and structure more medical knowledge to investigate the application of MKN in a variety of domains.

6.2 Inference

We plan to improve the arithmetic efficiency, identifying and exploiting the possibility of inference with huge amount of medical knowledge.

6.3 Learning

We plan to develop algorithms for learning and replace the pseudo-log-likelihood function, study dynamic approaches to weight learning, and build MKNs from sparse data and incomplete data.

Acknowledgements

This work was supported by the Key Program of National Natural Science Foundation of China under grant no. NSFC71531007. The Chinese Electronic Medical Records and the Blood Examination Records used in this paper were provided by Second Affiliated Hospital of Harbin Medical University and XingYi People's Hospital, respectively.

References

[1] World Health Organization (WHO), "Diabetes Programme". [Online] Available: http://www.who.int/diabetes/en/.
[2] World Health Organization (WHO), "Cardiovascular Disease". [Online] Available: http://www.who.int/cardiovascular_diseases/en/.
[3] World Health Organization (WHO), "Cancer". [Online] Available: http://www.who.int/cancer/en/.
[4] A.T. Azar, S.M. El-Metwally, Decision tree classifiers for automated medical diagnosis, Neural Comput. Appl. 23 (7-8) (2013) 2387–2403.
[5] D. Lavanya, K.U. Rani, Ensemble decision tree classifier for breast cancer data, Int. J. Inf. Technol. Converg. Serv. 2 (1) (2012) 17.
[6] Y.C.T.Bo. Jin, Support vector machines with genetic fuzzy feature transformation for biomedical data classification, Inf. Sci. 177 (2) (2007) 476–489.
[7] M. Peker, A decision support system to improve medical diagnosis using a combination of k-medoids clustering based attribute weighting and SVM, J. Med. Syst. 40 (5) (2016) 1–16.
[8] D. Vassis, B.A. Kampouraki, P. Belsis, et al., Using neural networks and SVMs for automatic medical diagnosis: a comprehensive review, Int. Conf. Integrated Inf. 1644 (1) (2015) 32–36.
[9] Y.Y. Wee, W.P. Cheah, S.C. Tan, et al., A method for root cause analysis with a Bayesian belief network and fuzzy cognitive map, Expert Syst. Appl. 42 (1) (2015) 468–487.
[10] A.C. Constantinou, N. Fenton, W. Marsh, et al., From complex questionnaire and interviewing data to intelligent Bayesian network models for medical decision support, Artif. Intell. Med. 67 (1) (2016) 75–93.
[11] Y. Huang, P. McCullagh, N. Black, R. Harper, Feature selection and classification model construction on type 2 diabetic patients' data, Artif. Intell. Med. 41 (3) (2007) 251–262.
[12] X. Liu, R. Lu, J. Ma, et al., Privacy-preserving patient-centric clinical decision support system on naive Bayesian classification, IEEE J. Biomed. Health Inform. 20 (2) (2016) 655–668.
[13] A. Bhardwaj, A. Tiwari, Breast cancer diagnosis using genetically optimized neural network model, Expert Syst. Appl. 42 (1) (2015) 4611–4620.
[14] F. Amato, A. López, E.M. Peña-Méndez, et al., Artificial neural networks in medical diagnosis, J. Appl. Biomed. 11 (2) (2013) 47–58.
[15] S. Palaniappan, R. Awang, Intelligent heart disease prediction system using data mining techniques, in: IEEE/ACS International Conference on Computer Systems and Applications, 2008, pp. 108–115.
[16] J.A. Sanz, M. Galar, A. Jurio, et al., Medical diagnosis of cardiovascular diseases using an interval-valued fuzzy rule-based classification system, Appl. Soft Comput. 20 (1) (2014) 103–111.
[17] N.A. Korenevskiy, Application of fuzzy logic for decision-making in medical expert systems, Biomed. Eng. 49 (1) (2015) 46–49.
[18] C.A. Pena-Reyes, M. Sipper, Evolutionary computation in medicine: an overview, Artif. Intell. Med. 19 (1) (2000) 1–23.
[19] A. Jain, Medical diagnosis using soft computing techniques: a review, Int. J. Artif. Intell. Knowl. Discov. 5 (3) (2015) 11–17.
[20] W.P. Goh, X. Tao, J. Zhang, et al., Decision support systems for adoption in dental clinics: a survey, Knowl. Based Syst. 104 (1) (2016) 195–206.
[21] J Wang, M Domingos P, Hybrid Markov logic networks, in: AAAI, 8, 2008, pp. 1106–1111.
[22] M. Richardson, P. Domingos, Markov logic networks, Mach. Learn. 62 (1-2) (2006) 107–136.
[23] D Lowd, P Domingos, Efficient weight learning for Markov logic networks, in: European Conference on Principles of Data Mining and Knowledge Discovery, Springer, Berlin Heidelberg, 2007, pp. 200–211.
[24] H Poon, P Domingos, Sound and efficient inference with probabilistic and deterministic dependencies, in: AAAI, 6, 2006, pp. 458–463.
[25] A Artikis, O Etzion, Z Feldman, et al., Event processing under uncertainty, in: Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems, ACM, 2012, pp. 32–43.
[26] B Gutmann, M Jaeger, L. De Raedt, Extending ProbLog with continuous distributions, in: International Conference on Inductive Logic Programming, Springer, Berlin Heidelberg, 2010, pp. 76–91.
[27] PRIMARY PSYCHIATRY, "Electronic Medical Records." [Online] Available: http://primarypsychiatry.com/electronic-medical-records/.
[28] A.K. Sari, W. Rahayu, M. Bhatt, Archetype sub-ontology: improving constraint-based clinical knowledge model in electronic health records, Knowl. Based Syst. 26 (1) (2012) 75–85.
[29] S.L. Ting, S.K. Kwok, A.H.C. Tsang, et al., A hybrid knowledge-based approach to supporting the medical prescription for general practitioners: real case in a Hong Kong medical center, Knowl. Based Syst. 24 (3) (2011) 444–456.
[30] Ö. Uzuner, Y. Luo, P. Szolovits, Evaluating the state-of-the-art in automatic de-identification, J. Am. Med. Inform. Assoc. 14 (5) (2007) 550–563.
[31] J. Kazama, K. Torisawa, Exploiting Wikipedia as external knowledge for named entity recognition, in: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007, pp. 698–707.
[32] M. Song, W.C. Kim, D. Lee, et al., PKDE4J: entity and relation extraction for public knowledge discovery, J. Biomed. Inform. 57 (1) (2015) 320–332.
[33] I2B2, "Informatics for Integrating Biology & the Bedside." [Online] Avaiable: https://www.i2b2.org/.
[34] WILAB-HIT, "Resources." [Online] Avaiable: https://github.com/WILAB-HIT/Resources/.
[35] B He, B Dong, Y Guan, et al., Building a comprehensive syntactic and semantic corpus of Chinese clinical texts, J. Biomed. Inform. (2017).
[36] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, 2014.
[37] E. Ising, Beitrag zur theorie des ferromagnetismus, Zeitschrift für Physik A Hadrons and Nuclei 31 (1) (1925) 253–258.
[38] G.E. Hinton, T.J. Sejnowski, Optimal perceptual inference, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1983, pp. 448–453.
[39] E. Yilmaz, E. Kanoulas, J.A. Aslam, A simple and efficient sampling method for estimating AP and NDCG, in: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2008.
[40] Project Tuffy, "Meet Tuffy." [Online] Avaiable: http://i.stanford.edu/hazy/tuffy/.

Publication 3

De-identification of medical records using conditional random fields and long short-term memory networks

Zhipeng Jiang, Chao Zhao*, Bin He, Yi Guan, and Jingchi Jiang*

Journal of Biomedical Informatics 75 (2017) S43–S53

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

De-identification of medical records using conditional random fields and long short-term memory networks

Zhipeng Jiang[1], Chao Zhao[1], Bin He, Yi Guan*, Jingchi Jiang

School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China

ARTICLE INFO

Keywords:
Protected health information
De-identification
Conditional random fields
Long short-term memory networks

ABSTRACT

The CEGS N-GRID 2016 Shared Task 1 in Clinical Natural Language Processing focuses on the de-identification of psychiatric evaluation records. This paper describes two participating systems of our team, based on conditional random fields (CRFs) and long short-term memory networks (LSTMs). A pre-processing module was introduced for sentence detection and tokenization before de-identification. For CRFs, manually extracted rich features were utilized to train the model. For LSTMs, a character-level bi-directional LSTM network was applied to represent tokens and classify tags for each token, following which a decoding layer was stacked to decode the most probable protected health information (PHI) terms. The LSTM-based system attained an i2b2 strict micro-F_1 measure of 0.8986, which was higher than that of the CRF-based system.

1. Introduction

The Electronic Health Record (EHR) is the systematized collection of electronically stored health information of patients in digital format [1]. It consists of a large amount of medical knowledge, and is a novel and rich resource for clinical research. A limitation of the large-scale use of EHR is the privacy of information contained in the text. To protect the privacy of patients and medical institutions, the US Congress passed the Health Insurance Portability and Accountability Act (HIPAA) in 1996. HIPAA defines 18 kinds of protected health information (PHI) that must be removed before the EHR can be reused, such as names, all geographic subdivisions smaller than a State, and so on.[2] The i2b2 (Informatics for Integrating Biology and the Bedside) Center defines more types of PHI based on the HIPAA-PHI categories. The removal of PHI information from clinical narratives is called *de-identification*. However, manual de-identification is time consuming, expensive, and ineffective. To explore the possibility of automatic de-identification approaches using natural language processing (NLP), i2b2 held its first clinical narrative de-identification event in 2006 [2], and again in 2014 (The 2014 i2b2/UTHealth NLP Shared Task 1) [3], and 2016 (CEGS N-GRID 2016 Shared Task 1 in Clinical NLP) [4]. Most participants of the events proposed solutions to this problem using machine learning algorithms, whereas rule-based methods were also presented.

According to the results, the highest-ranking team attained an i2b2 strict micro-F_1[3] of over 0.9. Although certain PHI cannot be de-identified by an automatic system, studies have shown that it is sufficient for preventing re-identification from these processed records [5,6]. These studies together confirmed the efficiency of the automatic de-identification systems.

In this paper, we describe two de-identification systems utilized in CEGS N-GRID 2016 Shared Task 1 based on conditional random fields (CRFs) and long short-term memory networks (LSTMs). We also contrast the principle and performance of these two systems, and analyze the identification errors. The remainder of this paper is structured as follows: In Section 2, we give a brief introduction of recent models used for named entity recognition (NER) and medical narrative de-identification. Section 3 describes the general pipeline of this task, and Sections 4 and 5 provide details of the principles and implementation of CRFs and LSTMs, respectively. We then report the evaluation of our systems on the CEGS N-GRID 2016 Shared Task 1 dataset in Section 6, and provide results and discussion in Sections 7 and 8, respectively. Our conclusions and directions for future studies are presented in Section 9.

2. Related work

From the perspective of NLP, de-identification is an NER task. NER

* Corresponding author.
E-mail addresses: hit.jiang@hotmail.com (Z. Jiang), zhaochaocs@gmail.com (C. Zhao), hebin_hit@hotmail.com (B. He), guanyi@hit.edu.cn (Y. Guan), jiangjingchi0118@163.com (J. Jiang).

[1] These authors contributed equally to this work.

[2] https://www.hipaa.com/hipaa-protected-health-information-what-does-phi-include/ lists all PHIs in the HIPAA.

[3] The evaluation measures are introduced in Section 6.4.

http://dx.doi.org/10.1016/j.jbi.2017.10.003
Received 31 January 2017; Received in revised form 30 September 2017; Accepted 3 October 2017
Available online 13 October 2017

Table 1
Regular expressions used for tokenization.

Regular expression	Original token	After tokenization	Comment
[A-Za-z][0-9]	a26 yo man	a 26 yo man	Digit
[0-9][A-Za-z][A-Za-z]+	10/6/2098SOS	10/6/2098 SOS	
[A-Z]{3,}[a-z]{2,}+	USMeaningful	US Meaningful	Uppercase
[a-z][A-Z]	WhalenChief	Whalen Chief	
\d{1,2}([/-])(\d{1,2}(\1))?\d{2,4}	09/14/2067CPT	09/14/2067 CPT	DATE
\D\d{3}\D{0,2}\d{3}\D{0,2}\d{4}	109 121 1400Prior	109 121 1400 Prior	PHONE
\w+@\w+\.[a-z]+	hcuutaj@ bdd.comOther	hcuutaj@ bdd.com Other	EMAIL

was first introduced at the Sixth Message Understanding Conference (MUC-6) [7], and has developed rapidly in the 20 years since. Many statistical learning algorithms have been applied to it, such as hidden Markov networks (HMMs) [8], CRFs [9], support vector machines (SVMs) [10]. These methods all depend on several features and regard the problem as a tagging process, which is the classification of each token over text sequences. The common features include word-level, list-level (i.e., dictionary features), and document-level features. The CRF is the most commonly used model in general NER tasks because of its theoretical advantage and experimental efficiency [11]. In the past two i2b2 de-identification tasks in 2006 and 2014, the best systems were based on CRFs. In the 2014 task, [12] identified the word-token, context, orthographical, sentence-level, and dictionary features to train a CRF model, and achieved the highest F-measure of 0.936 of all participants. Moreover, manually derived post-processing approaches are often used, and can yield considerable improvement in some PHI categories, like DATE and HOSPITAL.

Despite their good performance, a problem with these approaches is that performance is highly dependent on the extracted features. The quality of the features relies heavily on the experience of researchers and their familiarity with the data. In recent years, the rise of representation learning [13] methods has brought new vitality to NER tasks. Representation learning attempts to extract efficient features directly from data, and then applies deep neural networks [14], such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to compose the features. Features are then transformed layer by layer using non-linear functions to fit the intricate structures of the data. Better performance is obtained than that by CRFs when the training data is abundant. Many modified and combined deep neural networks have been applied to tagging tasks, from the simplest feed-forward neural networks [15] to long short-term memory (LSTM) networks [16] and various combinations, such as LSTM-CRF [17], LSTM-CNNs [18], CNN-LSTM-CRF [19], and so on. Lample et al. [20] proposed a character-level bidirectional LSTM-CRF architecture and claimed to obtain state-of-the-art NER results in standard evaluation settings. Dernoncourt et al. [21] transferred this work to EHR de-identification tasks and obtained an F_1 measure of 0.9785 on the i2b2 2014 dataset, higher than that of the best CRF-based approach in [12] (0.936).

In addition to statistical approaches and deep learning methods, rule-based methods are helpful for NER, although they are usually adopted as a deliberately weakened component in many academic papers [22]. Such rules include regular expressions, domain dictionaries, and a series of hand-crafted grammatical, syntactic, and orthographic patterns.

3. De-identification pipeline

Although many machine learning approaches are available for NER, they follow the general processing procedure of pre-processing, tagging, and post-processing, which is also used in our two de-identification systems. Pre-processing is indispensable when the data are not as clean as expected. After pre-processing, we use an algorithm to tag the entities through annotated data,[4] or use several models to boost the results. During post-processing, hand-derived rules are applied to correct potential tagging errors and find more missing entities. It is not indispensable, but can help improve the accuracy of the system in many cases. Since we had limitations of time during the task, we did not introduce a post-processing module. This section gives an overview of pre-processing and tagging modules in our two systems.

3.1. Pre-processing

In pre-processing, we focus on tokenization and sentence detection. Tokenization is necessary because some separators between entity tokens and ordinary tokens may be missing. Without these segments, the tagger cannot accurately detect the boundaries of entities. For example, in the phrase “09/14/2067CPT Code,” “09/14/2067” is the date entity, but “CPT Code” is not. Without tokenization, the tagger would either recognize “09/14/2067CPT” as an entity or not, and neither is correct. In this case, extent or missing errors[5] occur.

Some kinds of cases can be tokenized easily through regular expressions, such as the above example. In other cases, a token dictionary is needed. For simplicity, we only tokenize text by regular expressions listed in Table 1.

Sentence detection is another part of pre-processing. Records must be separated as sentences to feed into models because neither CRFs nor LSTMs can receive sentences that are very long. Detection only according to punctuation can cause problems. For example, if “Dr. Vincent” is separated according to the period, “Vincent” becomes the first word of the sentence and “Dr.,” which is an important identifier of the DOCTOR category is lost. Sentence boundaries can be detected using either rule-based methods or machine learning-based methods [23]. In this study, we detected the boundaries using the OpenNLP sentence detector,[6] which is a supervised toolkit. We only used the officially provided detection model.

3.2. Tagging

We used the BIOEU tagging schema for this task. It tagged the **B**eginning, **I**ntermediate, and **E**nd parts of the entities, as well as the **O**utside of a named entity. If an entity consisted of only one token, it was simply tagged as of **U**nit length.

Like other classification problems, tagging relies on feature extraction. The traditional features of text are indicator functions. These features are really flexible and have been shown to be efficient. However, they face two problems. First, these features are handcrafted, and feature templates need to be re-designed when handling new data. Second, these features are numerous and sparse. This large number of features leads to more parameters and higher computational cost. It

[4] In this paper, we focus on supervised NER approaches.
[5] We discuss the type of errors in Section 8.3.
[6] https://opennlp.apache.org/documentation/1.5.3/manual/opennlp.html#tools.sentdetect.

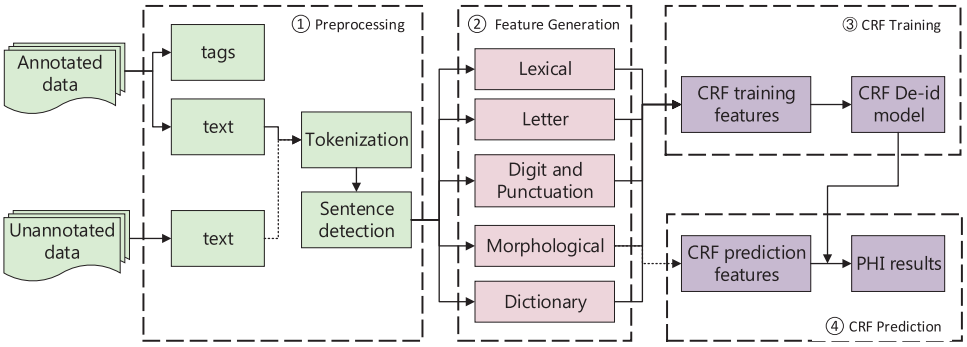


Fig. 1. The pipeline of CRF-based tagging system. The raw text is parsed from training data, tokenized, and split into sentences. Five categories of features are generated for each token. These features are then fed into CRF++ toolkit to obtain the de-identification tagging model. The same pre-processing and feature generation steps are applied to test data, then the trained model would label each test token according to their features. The labels are decoded and the system would output the final PHI results.

also restricts the size of the context windows during feature extraction.

The emergence of representation learning and deep neural networks led to the introduction of new feature extraction approaches. Researchers tried to learn the distributed representation of each token, called word vectors, or word embedding [24]. This kind of representation maps discrete words as vectors in a continuous low-dimensional vector space. That is to say, for each word w_i, there is a map $\phi(\cdot)$ to ensure that $\phi(w_i) = \mathbf{w}_i \in \mathbb{R}^{d_e}$, where d_e is the number of dimensions of the embedding. The word embeddings are trained according to the contexts of the words, and words with similar semantic meanings are mapped into nearby representations. In this way, language is converted into signals that can be numeralized and computed, just like sounds and images. The vectors of each word can be directly regarded as features, and d_e is usually between 50 and 1000, which is much smaller than in one-hot representation. The difference between word embeddings and other signals is that the operations for these vectors, such as translation, rotation, scaling and superposition, have no actual meanings, or their meanings are still unknown. However, to combine the representation of words into higher-level representations, like the representations of sentences, the word embeddings must be manipulated. Thus, different kinds of neural networks are used to combine word embeddings, including CNNs, RNNs, and recursive neural networks.

4. CRF-based model

The pipeline of our CRF-based system is shown in Fig. 1. The CRF model was implemented using the CRF++ toolkit.[7] The details of feature generation is given below, while the representation and inference part of CRFs can be found in Appendix B.

A large number of rich features were extracted to feed the CRF classifier, including lexical, orthographic, morphological, and dictionary features. All features of the given token within its ±2 context window were considered. More details of the features are listed in Table 2. The POS and chunk features were obtained by utilizing OpenNLP POS tagger and chunker.[8] The dictionaries used in the system were collected from the training data as well as Wikipedia.

It is not the case that the more features, the better the performance. Some features may introduce noise and degrade performance. To avoid bad features, we selected them in a greedy way. That is, we added features in turn and evaluated the tagging results. Once performance degraded, we discarded the given added feature. This approach cannot guarantee optimal feature selection, but can reduce time complexity from $O(2^n)$ to $O(n)$ with ease. The contribution of each feature subcategory to the final performance is analyzed in Section 7.

5. LSTM-based model

LSTM is a special type of RNN. It utilizes word embeddings as inputs. The embedding of the given tokens are then combined with the context embeddings by the LSTM layer, which yields the new hidden representation of the tokens. Finally, the hidden representations are used directly for classification. These three steps are discussed in the following subsections. Fig. 2 shows the architecture of the LSTM networks used in the task.

5.1. Long short-term memory networks

RNN is one way to combine a sequence of word embeddings $\mathbf{x}_{<1:t>}$ to an embedding . The combination is defined using the recurrent formula

$$\mathbf{h}_t = \tanh(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b}), \quad (1)$$

where $\mathbf{x}_t \in \mathbb{R}^{d_e}$ is the word embedding of t-th word, $\mathbf{W} \in \mathbb{R}^{d_h \times d_e}$,$\mathbf{U} \in \mathbb{R}^{d_h \times d_h}$,$\mathbf{b} \in \mathbb{R}^{d_h}$ are the weight and bias parameters to be learned, and the initial condition is $\mathbf{h}_0 = \mathbf{0}$.

After feeding all $\mathbf{x}_{<1:t>}$ into the above formula, we obtain $\mathbf{h}_t$, which contains not only the given token, but also the previous context as well, and can be used as a new representation of t-th word w_t. However, this kind of combination leads to difficulty in gradient descent while training parameters, since the partial derivatives of $\mathbf{h}_t$ with respect to $\mathbf{h}_i$ continuously increase or decrease with growth in $t-i$, and finally vanish or explode [25,26]. To address this problem, LSTMs are proposed [16]. They limit the increase in $t-i$ through a forget gate $\mathbf{f}_t$. In this study, we use a modified LSTM [27], which sets the forget gate $\mathbf{f}_t = \mathbf{1}-\mathbf{i}_t$ to reduce the parameters. The details of the gate and the cell computation of LSTM can be found in Appendix C.

The LSTM can combine the embeddings forward as well as backward. Once we obtain the forward hidden layer $\overrightarrow{\mathbf{h}_t}$ and the backward hidden layer $\overleftarrow{\mathbf{h}_t}$ and concatenate them together as $\overleftrightarrow{\mathbf{h}_t}$, we get the contextual information for this token to some extent.

5.2. Word representation enhancement

As the input of LSTM layer, the word embeddings of each token are pre-trained from the training data. We utilized the Word2Vec toolkit[9] and selected the skip-gram model to obtain the pre-trained, non-case-sensitive word embeddings.

Since we only used the training data, we were not able to obtain the representations of tokens that did not appear in the training set. These tokens are called out-of-vocabulary (OOV) tokens. One simple solution to this problem is to assign the embeddings of OOV tokens randomly. A better solution is to obtain the representations according to the characters composing the words [28]. Although there are hundreds of thousands of tokens in the English corpus, it consists of less than 100

[7] https://taku910.github.io/crfpp/.
[8] https://opennlp.apache.org/.
[9] https://code.google.com/archive/p/word2vec/.

Table 2
The features utilized at the CRFs-based system.

Category	Features	Feature instantiations of "Vincent"
Lexical	Lowercase	Vincent
	Word lemma	Vincent
	POS tag of the token	NNP
	Chunk tag	I-NP
	Long Shape of the token	Aaaaaaa
	Length of the token	7
Letter	Whether the token contains a letter	1
	Whether the token contains a capital letter	1
	Whether the token begins with a capital letter	1
	Whether all characters in the token are capital letters	0
Digit and punctuation	Whether the token contains a digit	0
	Whether all characters in the token are digits	0
	Whether the token contains a punctuation character	0
	Whether the token consists of letters and digits	0
	Whether the token consists of digits and punctuation characters	0
Morphological	First two characters of the token	Vi
	Last two characters of the token	nt
	First three characters of the token	Vin
	Last three characters of the token	ent
	First four characters of the token	Vinc
	Last four characters of the token	cent
Dictionary	Whether the lowercase of the token is in the "profession" dictionary	0
	Whether the lowercase of the token is in the "city" dictionary	0
	Whether the lowercase of the token is in the "country" dictionary	1
	Whether the lowercase of the token is in the "state" dictionary	0

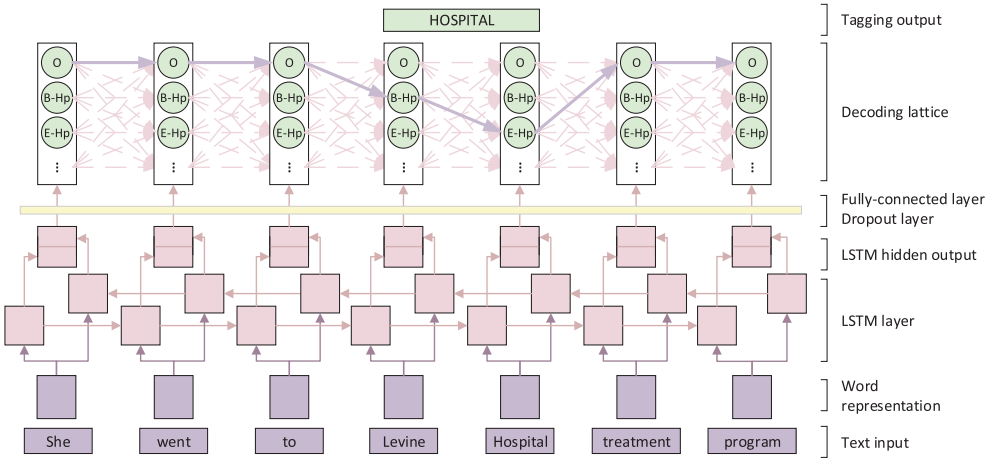


Fig. 2. The architecture of LSTMs tagging model. The word-level LSTMs receive the representation of each token as input and provide the hidden layer $\overleftrightarrow{\mathbf{h}}_t$ as output. This output is used to predict the probability of tags of each word x_t through a fully-connected layer, where the dropout is applied. In the decoding lattice, we find the most probable tag path from the tag lattice, along which the PHI information could be obtained.

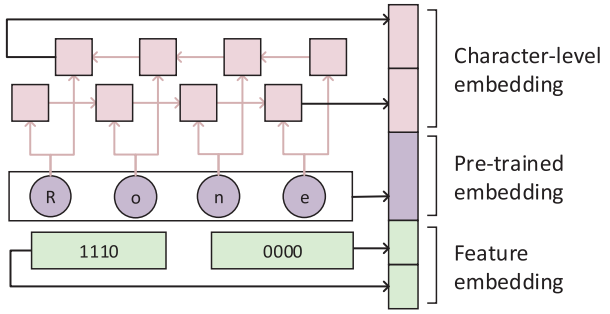


Fig. 3. An example of the enhanced representation of word "Rone". It is composed of the character-level embedding, the pre-trained embedding and the feature embedding.

characters. If we regard each token as a sequence of characters, LSTMs can be used to obtain two hidden representations of it. After concatenating the hidden representations, we get the character-level word embedding, which can represent the morphological meanings of the OOV token to some extent. Because we regard the upper and lowercase of one letter as different characters, the character-level word embeddings are case-sensitive.

Moreover, although the word embeddings can be utilized directly as features, incorporating hand-crafted features is also helpful, for example, the capital and dictionary features listed in Table 2. We used two four-bit binary numbers to indicate the capital and the dictionary feature values of each word, and allocated two feature embeddings accordingly. For example, the word "Vincent" in Table 2 can be encoded as "1110" and "0010", and these two codes can be further represented as two feature embeddings.

Based on the considerations above, we concatenated the character-level word embeddings and feature embeddings to the pre-trained word embeddings to enhance the word representation. Fig. 3 gives an example of the enhanced word representation. The contribution of each part of the enhanced representation is verified in Section 7.

5.3. Label decoding

Once we get hidden representation $\overleftrightarrow{\mathbf{h}}_t$ of each token w_t and its

context, we can predict its corresponding label immediately. That is,

$$P_t = \text{softmax}(\mathbf{W}_l \overleftrightarrow{\mathbf{h}}_t + \mathbf{b}_l) \tag{2}$$

$P_t \in \mathbb{R}^{d_l}$ and the i-th entry of P_t is the probability that w_t is labeled as the i-th tag. d_l is the number of candidate tags. $\mathbf{W}_l \in \mathbb{R}^{d_l \times 2d_h}$ and $\mathbf{b}_l \in \mathbb{R}^{d_l}$ are the weight and bias parameters to be learned.

However, this classifier ignores the dependency of labels, which is helpful for tagging. We model this dependency in the classification layer by adding a transition matrix $\mathbf{M} \in \mathbb{R}^{d_t \times d_t}$, which is depicted as the "decoding lattice" in Fig. 2. $\mathbf{M}_{ij}$ is the unnormalized transition score from the i-th tag to the j-th tag. The transition score is time invariant, which means that it is independent of the tokens. In this way, the score of a sequence X and one of its predictions y^i is a combination of the classification score and the transition score:

$$s(X, y^i) = \sum_{t=0}^{T-1} \mathbf{M}_{y_t^i y_{t+1}^i} + \sum_{t=0}^{T} P_{t, y_t^i} \tag{3}$$

The probability of the j-th prediction y^i can be obtained via a softmax classifier

$$p(y^i|X) = \frac{e^{s(X, y^i)}}{\sum_i e^{s(X, y^i)}} \tag{4}$$

During training, the log-probability of the correct label sequence is maximized:

$$\log(p(y^i|X)) = s(X, y^i) - \log(\sum_i e^{s(X, y^i)}) \tag{5}$$

Although the number of possible numbers of y increases in $O(n^T)$, the calculation of Eq. (5) can be completed in $O(n^2)$ time using dynamic programming. The details of the derivation can be found in Appendix D.

In the decoding, the label sequence that obtains the maximum score is selected as the prediction:

$$y^* = \arg\max_i s(X, y^i) \tag{6}$$

6. Experiment

6.1. Corpus

The medical record set used in CEGS N-GRID 2016 Shared Task 1 in Clinical NLP contained 1000 psychiatric evaluation records provided by Harvard Medical School. They consisted of XML documents containing raw text and PHI annotations. Table 3 shows the statistics of some main measurements of the corpus.

6.2. Experimental setup

For CRF based system, we set the cut-off threshold of the features -f to 4, the hyper-parameter -c to 10, and used L2-norm for regularization. The hyper-parameter was determined via a subset of training data during the task. This subset is referred to from here on as the "validation set".

For LSTM-based system, the dimensions of each layer were set as follows:

- Character embedding dimension: 25
- Character-level LSTM hidden layer: 25
- Pre-trained embedding dimension: 100
- Capital embedding dimension: 6
- Dictionary embedding dimension: 6
- Word-level LSTM hidden layer: 64

Dropout [29] with a probability of 0.5 was applied to prevent overfitting. The model was trained using the stochastic gradient descent (SGD) algorithm, and the learning rate was set to 0.005. The dimension of pre-trained embedding was tuned via the validation set, while other parameters were determined through experience. The system should perform better after fine-tuning.

Table 3
Statistical overview of CEGS N-GRID 2016 Shared Task 1 corpus. The number of tokens and vocabularies were counted after pre-processing.

	#Notes	#Tokens	#PHI	#Unique PHI	#Vocabulary
Train	600	1,432,251	20,845	14,120	28,308
Test	400	956,168	13,519	9301	23,591

6.3. Improvement following best official run

We submitted three outputs to the task committee, the results of which are referred to from here on as the "official runs". In the experiments after the challenge, we enhanced the results further by improving the tagging models and the pre-processing module. This yielded two unofficial runs for CRFs and LSTMs.

The improvements to the models were centered around feature selection and hyper-parameter tuning. For the CRFs, we enriched the features and re-selected them in a greedy way. We also fine-tuned the hyper-parameters "-c" of the CRF++ toolkit to balance overfitting and underfitting. For LSTMs, we set the number of dimensions of the word embeddings from 50 to 100. We also added more training epochs to ensure that the parameters had been trained sufficiently.

More importantly, we added sentence detection and further improved tokenization in the pre-processing module. In the official run, we simply detected sentences according to punctuation, and many integrated sentences were hence separated, leading to a loss of contextual information. Besides, we did not tokenize the corpus in the official run as thoroughly as described in Table 1, because we thought that excessive tokenization might have split a complete PHI term into two and caused errors. In the unofficial run, we introduced the sentence detection strategy, and enriched regular expressions to ensure that as many potential entities as possible were well tokenized. The result showed that the side-effect of excessive tokenization was not severe.

The contribution of these improvements to the final performance is discussed in Section 7.

6.4. Evaluation

The system output was evaluated using precision (P), recall (R), and the F_1 measure, which are defined as follows:

$$P = \frac{\#(\text{true positives})}{\#(\text{true positives} + \text{false positives})} \tag{7}$$

$$R = \frac{\#(\text{true positives})}{\#(\text{true positives} + \text{false negatives})} \tag{8}$$

$$F_1 = \frac{2PR}{P + R} \tag{9}$$

According to the count method of true positives, the evaluation measures can further be classified in three independent dimensions.

There were two sets of PHI categories in the evaluation, defined by i2b2 and HIPAA. The i2b2 PHI categories were an expanded set of the HIPAA categories and contained more PHI terms, such as PROFESSION and COUNTRY.

- When the system was evaluated according to **i2b2 categories**, all subcategories under the seven main ones were evaluated.
- When the system was evaluated according to **HIPAA categories**, only the categories defined by HIPAA were evaluated.

Table 4
Evaluation scores for the best official run as well as two unofficial runs implemented using CRFs and LSTMs. *F* in the table refers to micro-F_1 measure. The highest strict measures are shown in bold, and measures that have statistical significance compared with those of other two runs are marked with ★.

		Best official run			Unofficial CRF run			Unofficial LSTM run		
		Strict	Relaxed	Token	Strict	Relaxed	Token	Strict	Relaxed	Token
I2B2	P	0.8418	0.8467	0.8881	**0.923**	0.924	0.9463	0.9229	0.9240	0.9431
	R	0.8728	0.8778	0.9087	0.8411	0.842	0.87	**0.8755**	0.8765	0.9015
	F	0.857	0.862	0.8983	0.8802	0.8811	0.9065	**0.8986★**	0.8996	0.9218
HIPAA	P	0.8767	0.8818	0.9125	0.9321	0.9333	0.9558	**0.9324**	0.9337	0.9509
	R	0.9001	0.9054	0.9271	0.8696	0.8708	0.8696	**0.8972**	0.8985	0.9198
	F	0.8882	0.8934	0.9197	0.8998	0.901	0.8998	**0.9145★**	0.9157	0.9351

The evaluations also differed according to whether we assessed the system at the instance level or the record level:

- **micro-F**: All of the PHI instances in the dataset were evaluated together.
- **macro-F**: Each record was evaluated and the ultimate score was obtained using the average.

If only part of the tokens of an integrated entity were identified by the system, entity-level and token-level evaluations can lead to different measures. In entity-level evaluation, there were two standards according to different matching strictness:

- **Entity level**: An entity must be identified as a whole, despite the number of tokens it contained.
  - **Strict**: The recognized entity must exactly match the first and last offsets of the gold standard entity.
  - **Relaxed**: The last offset can be off by up to 2.
- **Token level**: If the entity was separated into several parts and each was identified as the correct type, the entity was regarded as correctly identified. For example, if the PHI "2072 winter" in golden-standard with the category DATE is annotated as two DATE terms "2072" and "winter", the token-level evaluation would regard the output as correct, but the entity level would not.

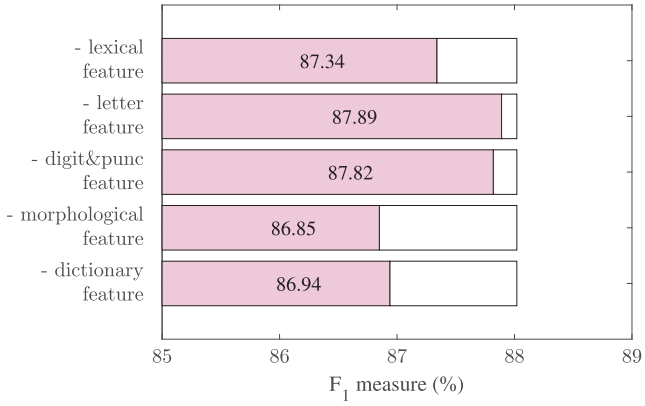


Fig. 4. Overall performance curve of CRF-based system, with one feature sub-category removed from the feature set.

Unless otherwise specified, all F_1 measures in the remainder of this paper are strict entity-level micro-F_1 measure, the primary evaluation metric in CEGS N-GRID 2016 Shared Task 1.

7. Results

The best result of the three official runs mixed the outputs of the

Table 5
The evaluation measures for each sub-category. The table lists the number of PHI instances in the training data, the gold standard data, the system output, as well as the number of instances of agreement between the last two. It also lists the P, R, and F_1 values of each sub-category. Only the sub-categories that appear in the training data are listed.

Category	Sub-category	#Train	#Gold	#System	#Agree	P	R	F
Name	PATIENT	1270	837	658	597	0.9073	0.7133	0.7987
	DOCTOR	2396	1567	1587	1491	0.9395	0.9515	0.9455
	USERNAME	25	0	0	0	–	–	–
PROFESSION		1471	1010	828	688	0.8309	0.6812	0.7486
LOCATION	HOSPITAL	2196	1327	1211	1096	0.9050	0.8259	0.8637
	ORGANIZATION	1113	697	552	434	0.7862	0.6227	0.6950
	STREET	46	34	28	27	0.9643	0.7941	0.8710
	CITY	1394	820	892	748	0.8386	0.9122	0.8738
	STATE	662	481	459	419	0.9129	0.8711	0.8915
	COUNTRY	666	376	327	307	0.9388	0.8165	0.8734
	ZIP	23	17	17	17	1.0000	1.0000	1.0000
	LOCATION-OTHER	25	19	13	4	0.3077	0.2105	0.2500
AGE		3637	2354	2317	2234	0.9642	0.9490	0.9565
DATE		5723	3821	3790	3646	0.9620	0.9542	0.9581
CONTACT	PHONE	143	113	112	106	0.9464	0.9381	0.9422
	FAX	4	5	3	3	1.0000	0.6000	0.7500
	EMAIL	2	5	1	0	0	0	0
ID	URL	5	3	1	0	0	0	0
	MEDICALRECORD	4	2	1	0	0	0	0
	HEALTHPLAN	4	2	0	0	0	0	0
	LICENSE	38	21	28	19	0.6786	0.9048	0.7755
	IDNUM	2	8	0	0	0	0	0

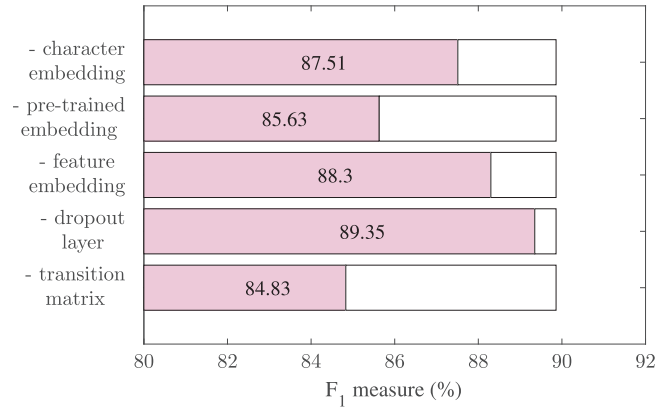


Fig. 5. Overall performance curve of LSTM-based system, with one layer removed from the LSTM architecture.

CRFs (F_1 = 0.845) and the LSTMs (F_1 = 0.861), and achieved an F_1 score of 0.857. The median F_1 score of all system outputs of the task participants was 0.822 (standard deviation = 0.183, mean = 0.779, minimum = 0.019). The best system achieved an F_1 of 0.914 [4]. With the improvements described in Section 6.3, the F_1 score of the CRF model increased by 3.52% and that of the LSTM model by 3.71%. Table 4 lists the evaluation measures of the best official run and the two unofficial runs based on the CRFs and the LSTMs. Details of the evaluation measures for each sub-category are listed in Table 5. Only the results of LSTM-based system are listed.

To verify the contribution of each module of system to overall performance, we removed them separately in turn, and re-calculated the F_1 measure. For the CRF model, we removed one feature sub-category of five. The results are shown in Fig. 4. For the LSTM model, we sequentially removed one of the three word representation parts. We also removed the dropout layer and the decoding lattice to show their influence on overall performance. When the decoding lattice was removed, the probability that the word w_t was labeled using tag t_i was calculated directly from Eq. (2). The results are shown in Fig. 5.

To demonstrate the effectiveness of the improvements described in Section 6.3, we calculated the statistical significance among the strict results of the three runs using approximate randomization [30,31], which has been used in the last two i2b2 challenges [2,3]. The best strict results with significant differences are marked with a star in Table 4. We further evaluated the individual contributions of improvements on model and pre-processing module to the increase of performance. We first modified the model and then improved the pre-processing module. The i2b2 micro-averaged P,R and F_1 measures were calculated after each step. The modification of the models increased the F_1 measures of the two systems by 1.8% and 1.68%, and the improvement in pre-processing increased the results by another 1.69% and 2.06%. Fig. 6 shows more details of the results, which suggest that pre-processing the raw text is highly beneficial before feeding it into the de-identification models.

8. Discussion

8.1. Discussion of the results

Generally speaking, the de-identification on the 2016 dataset is more difficult than that on 2014. One evidence is that the performance of the participating systems in the 2016 track was poorer than that in the 2014 track (maximum = 0.936, median = 0.845). Another is that we re-trained our participating system in 2014 track (F_1 = 0.923) [32] directly on the 2016 training set, and achieved a much lower F_1 score of 0.823 on the test set.

Our CRF-based system performed much worse on the official test set than the validation set, which drove the mixed result below that of LSTM only. This showed that the submitted CRF model, which was trained and validated by the training data, did not generalize to the test data well.

It is clear from Table 4 that the results of HIPAA were better than those of i2b2 because the systems performed poorly on some categories in the i2b2 set but not in the HIPAA set, such as PROFESSION, which reduced the F_1 score of the i2b2 categories. Further, the token-level evaluation measures were higher than the entity-level measures. It implied that the systems cannot identify the boundaries of PHI terms well in some cases, which led to extent errors.

For sub-categories in Table 5, the ZIP category obtained the highest F_1 score of 1.000 because of its highly regular form and relatively fixed context. AGE, DATE, DOCTOR, and PHONE also achieved F_1 scores of more than 0.9 for similar reasons. Although the PATIENT sub-category belonged to the categories NAME as well as DOCTOR, its recall rate was much lower, only slightly over 0.7. This was due to the high rate of missing errors, and potential causes are analyzed in Section 8.3.4. EMAIL and ID were also regular, but instances of these categories were rare compared with the above categories. A data-driven system was unlikely to learn patterns from so few instances, and thus yielded poor performance. For these categories, regular expressions may be helpful.

Three classes of sub-categories were shown in the LOCATION category. The first contained CITY, STATE, and COUNTRY, which had relatively fixed dictionaries, and obtained higher F_1 scores. The second contained HOSPITAL and STREET. These sub-categories had some signal words, such as “hospital,” “clinic,” “road,” and “avenue,” which were also helpful for recognition. The third class, which contained ORGANIZATION and LOCATION-OTHER, had neither complete

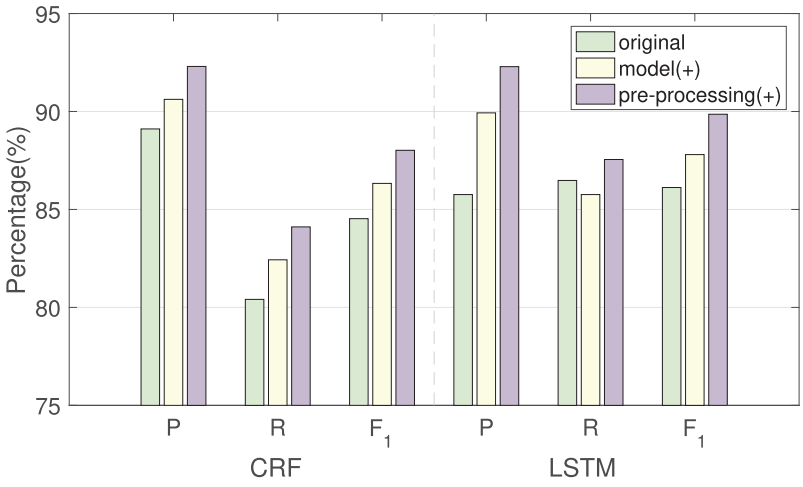


Fig. 6. P, R and F_1 measures change of two systems after the model and pre-processing module improvements. CRF-based system (left), LSTM-based system (right).

dictionaries nor signal words, and thus was more difficult to recognize.

PROFESSION was one of the most difficult categories to identify. One reason for this was the lack of dictionaries and signal words, and another was that the contexts for the terms in this category were complicated. Moreover, the terms in PROFESSION could be long, such as "Telecommunications Installation and Repair Worker" and "Inspector in Public and Environmental Health and Occupational Health and Safety." Identification of such terms needed to rely on longer contextual information, which is still a hard problem in NLP.

8.2. Comparison of CRFs and LSTMs

LSTMs outperform CRFs in all F_1 measures. LSTM-based systems have similar precision but much higher recall than CRF-based systems. This is because the hand-crafted features used in CRFs are selected carefully for token classification, but cannot cover all scenarios. On the contrary, the automatic features utilized by LSTMs are derived directly from data, and can depict intrinsic features hidden in the data. Thus, the features are more general and the recall is much higher than that of CRFs.

From a model-building perspective, LSTMs have an additional hidden layer compared with CRFs, which have only two layers. If we regard LSTMs as deep neural networks, CRF is a kind of shallow network. It has been claimed that deep neural networks have more powerful fitting capabilities compared with shallow log-linear models. Moreover, LSTM can model long-term dependency, which can capture contextual information for a longer period.

An advantage of CRFs is that they optimize entire sequences of tags rather than tags of each token. To use this in the LSTMs, we introduce the transition matrix while decoding the labels.

8.3. Error analysis

8.3.1. Error categories

Errors under the entity-level strict evaluation were divided into four categories according to [33]:

- Type error: The entity was identified by a correct start and end location but the wrong type.
- Extent error: The location span of the entity overlapped with that of a gold-standard entity, but did not match it exactly. There were three scenarios of overlapping:
  - Short: The location span of the entity fell within that of a gold-standard entity.
  - Long: The location span of the entity covered that of a gold-standard entity.
  - Short & Long: The location span of the entity neither fell into nor covered that of a gold-standard entity.
- Spurious error: The location span of the entity had no overlap with any correct entity.
- Missing error: The location span of the entity in the gold standard had no overlap with that of any entity in the system.

Fig. 7 shows the distribution of the four error categories. The percentages of type error, spurious error, and extent error were calculated based on the system output, whereas the percentages of missing error were calculated based on gold-standard data. If a PHI term produced by the system was incorrect, the error would certainly be grouped into one of the above four categories, except missing error. Consider the first column of Fig. 7 as an example. It shows that approximately 91% of the entities in PATIENT were identified by system according to the gold standard, 2% of which from the category DOCTOR, 4% were not PHI terms, and 1% were shorter than their corresponding standard answers. The sum of these percentages is not exactly 100% because the values in each block were rounded due to limitations of space. The original data for this figure can be found in Table A.1. From the first row of missing errors in the sub-figure, we see that approximately 23% of the terms in PATIENT in the gold-standard were not identified by the system.

8.3.2. Type errors

Type errors are shown as a confusion matrix in Fig. 7. It can be seen easily that entities tended to be identified incorrectly among the sub-categories belonging to a main category. For example, 19 entities of PATIENT were incorrectly identified as entities of DOCTOR, and 14 entities of DOCTOR were incorrectly identified as those of PATIENT. The same scenario also obtained among sub-categories in LOCATION

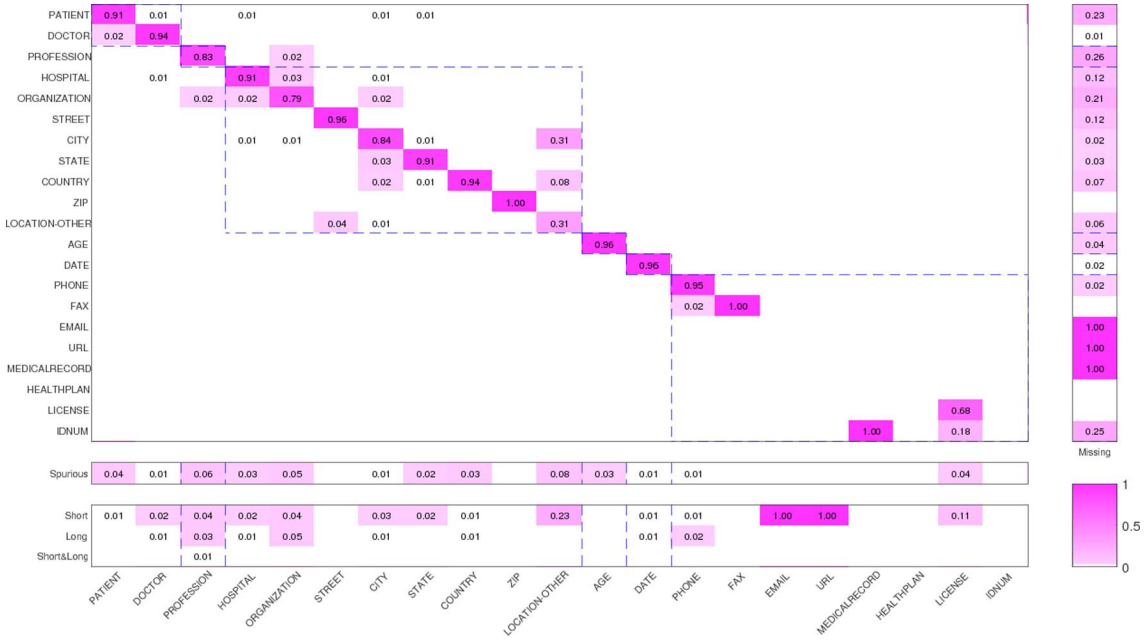


Fig. 7. The visualization of error distribution. The main sub-figure is a confusion matrix used to depict type error. The other three sub-figures show the distribution of missing, spurious, and extent errors. The meanings of each row and column are listed in the headings. Different PHI categories are separated by the dashed blue line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and CONTACT. The difficulty in finer-grained classification lay in the similarity of the morphological features and the context of these entities.

There were also several confusions between the categories PROFESSION and LOCATION. This was because they sometimes occurred together and shared similar contexts. For example, the sentence "a retired Landscape architect from Albemarle Corporation" contains a PROFESSION term (Landscape architecht) and an ORGANIZATION term (Albemarle Corporation).

8.3.3. Spurious errors

Spurious errors occurred when ordinary tokens had similar lexical or contextual features with real PHI entities. For example, a token with the first letter capitalized was likely to be identified as a PHI term, and one consisting of two digits tended to be recognized as DATE or AGE. In other cases, the word itself had more than one semantic meaning. For example, in the sentence "Her last depressive episode was last winter," the token "winter" was annotated as a DATE entity. However, in the phrase "winter boots," the token was just an adjective rather than a PHI term. There were also some cases where the tokens themselves were confusing. For example, the system identified "big company" and "JEwish day schools" as ORGANIZATION, though they were not exactly PHI terms.

8.3.4. Missing errors

Missing errors occurred more often in sub-categories with few instances in the training set. For EMAIL, URL, and MEDICALRECORD, there were fewer than 10 instances each, and it was hence natural that the error rate in the test set was high. In addition to CONTACT, PROFESSION and ORGANIZATION had the highest missing rates. As discussed above, this was due to a lack of complete domain dictionaries and signal words.

It is interesting that although both PATIENT and DOCTOR were sub-categories of NAME, the system yielded entirely different missing rate for these two sub-categories. The missing rate of entities in PATIENT was approximately 23%, much higher than the 1% of those in DOCTOR. In addition to the more complicated context of PATIENT, we found that the document frequency of PATIENT was much lower than that of DOCTOR. This meant that the system more easily remembered token of DOCTOR. The document frequency distributions of the main sub-categories are shown in Fig. 8.

8.3.5. Extent errors

The occurrence of extent errors showed that the model could not satisfactorily detect the boundaries of several entities. We checked cases for all three types of extent errors and described them below.

There were two main causes of short errors. The first was tokenization. For example, "Zenith Uni." was an ORGANIZATION entity. However, since the period was separated from "Uni" during tokenization, the system can only identify "Zenith Uni" as a PHI term. Another type of short error was that the system tags one entity as two. "State University of Wyoming" was an ORGANIZATION entity. However, the system tagged "State University" as an ORGANIZATION and "Wyoming" as a STATE. This showed that the system could not handle the "of" phrase structure well. It sometimes tagged two sub-structures of an integrated term with "of" as two independent entities. Similar examples were "winter of 2091" and "Cancer Center of America."

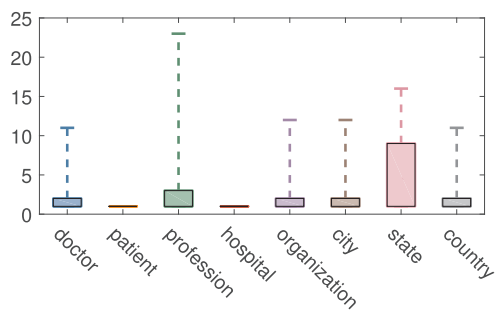


Fig. 8. The boxplot of document frequencies of the main sub-categories.

Similarly, these two problems can also cause long errors. For example, the system tagged "Educare-Fargo\," rather than "Educare-Fargo" as a "hospital" entity because of the failure of tokenizing "\" out from "Fargo." The system sometimes also tagged two or more entities as one. For example, the entity "computer science health informatic" identified by the system were in fact two entities in the gold standard. There were some long errors as well that were confusing. For example, the system tagged "**Woodland Park** High School" as an "organization," but the gold standard tagged only "Woodland Park" as a "city." Similar examples were "**landscaping** employer" and "**HMC** Home Services." Perhaps some extra information, like tf-idf, can help the system filter more general tokens, such as "high school" and "employer."

Short & Long errors occurred less frequently compared with the above two extent errors, and mainly in PROFESSION when preceded by an ORGANIZATION entity. For example, in the sentence "33yo married palauan female Bob Evans buildings construction worker," the gold standard tagged "Bob Evans buildings" and "construction worker" as ORGANIZATION and PROFESSION, respectively. However, the system tagged "buildings construction worker" as a PROFESSION entity.

8.4. Limitations

Text in the clinical domain has its own characteristics. The direct transformation of NER from the open to the clinical domain is not the best way. However, limited by the time available for the task, specific processing against clinical narratives was not introduced to the systems. We did not exploit the characteristics of the corpus fully, and therefore, no rule-based post-processing module was added. We also just utilized a small part of the dictionaries. The sentence detector and the POS/chunk tagger were open-domain toolkits. We believe these rules, resources, and domain-specific toolkits can further improve the performance of the system.

At the same time, open-domain prior knowledge is also helpful for some categories, such as PROFESSION and LOCATION, which are also common named entities in open-domain corpora. The pre-trained word embedding based on these large-scale corpora should be helpful for identification.

9. Conclusion

This paper proposed two automatic de-identification systems based on CRFs and LSTMs. The LSTM-based system attained a micro-F_1 score of 0.8986 in i2b2 strict evaluation, which was higher than that of the CRF-based system. LSTMs can identify PHI terms without depending on hand-crafted features and obtain higher recall rates than CRFs. In addition to the model, the pre-processing module can significantly affect the performance of the system. Accurate sentence detection and tokenization is a premise and foundation of subsequent PHI term recognition.

Furthermore, as Section 8.4 pointed out, we will attempt to incorporate prior knowledge and domain-specific resources to help increase the results for categories that yielded poor performance.

Conflict of Interest Statement

We declare that we have no conflict of interests.

Acknowledgment

TheCEGS N-GRID 2016 Shared Task 1 in Clinical Natural Language Processing was supported by NIH P50 MH106933, NIH

4R13LM011411. This work is also supported by the Natural Science Foundation of China (No. 71531007). The authors would like to thank the organizing committee for this task and the annotators of the dataset. We also thank the anonymous reviewers for their comments, which provided us with significant guidance.

Appendix A

Table A.1 is a quantitative version of Fig. 7.

Table A.1
Error distribution of system output.

	System output																					
	Pt	Dct	Pf	Hpt	Og	Strt	Ct	Stat	Ct	Zip	L-O	Age	Dt	Phn	Fax	Em	Url	Mrd	Hp	Lcs	ID	Missing	Total
PATIENT	597	19	1	8	2		5	6	1													187	826
DOCTOR	14	1491		4	2		4						1									22	1538
PROFESSION	1		688		9																	242	940
HOSPITAL	3	11		1096	19		13	2														151	1295
ORGANIZATION	2		18	26	434		14	1	1			1										136	633
STREET		1				27	1															4	33
CITY	1	1	2	11	5		748	4	1		4											18	795
STATE		2					30	419	1			1	3									14	470
COUNTRY	3			1	2		22	5	307		1											26	367
ZIP										17												0	17
LOCATION-OTHER				1	1	1	10				4											1	18
AGE								2				2234	5									97	2338
DATE	2	1	1		2		1	1				11	3646									56	3721
PHONE				1										106								2	109
FAX														2	3							0	5
EMAIL																0						4	4
URL																	0					2	2
MEDICALRECORD																		0				2	2
HEALTHPLAN																			0			0	0
LICENSE																				19		0	19
IDNUM																		1		5	0	2	8
Total	623	1526	710	1148	476	28	848	440	311	17	9	2247	3655	108	3	0	0	1	0	24	0	966	13,140
Spurious	26	20	52	33	29		9	8	11		1	59	45									1	294
Short	9	27	35	21	21		26	10	3		3	6	52	1		1	1			3		1	220
Long		14	26	9	26		9	1	2			5	37	2									131
sl			5										1									5	11
Total	9	41	66	30	47	0	35	11	5	0	3	11	90	3	0	1	1	0	0	3	0	6	362
System	658	1587	828	1211	552	28	892	459	327	17	13	2317	3790	112	3	1	1	1	0	28	0	973	

Appendix B

When the CRF is applied to NER, $X = \mathbf{x}_0,\mathbf{x}_1,...,\mathbf{x}_t,...,\mathbf{x}_T$ denotes the features of an input sentence of length T, where $\mathbf{x}_t$ is the features of the word x_t at position t and its context. And $Y = y_1,y_2,...,y_t,...,y_T$ denotes the corresponding output labels of each word x_t.

The feature set used in linear-chain CRF can be written as $\mathscr{F} = \{f_k(y_t,y_{t-1},\mathbf{x}_t)|\ \forall\ k\}$, where $f_k(y_t,y_{t-1},\mathbf{x}_t) = \boldsymbol{q}_k(\mathbf{x}_t)\mathbf{I}(y_t = y)\mathbf{I}(y_{t-1} = y')$. $\boldsymbol{q}_k(\mathbf{x}_t)$ is the k-th observed feature of x_t and its context, and $\mathbf{I}(\cdot)$ is the indicator function. It can further simplified as two kinds of features: $f_k(y_t,\mathbf{x}_t) = \boldsymbol{q}_k(\mathbf{x}_t)\mathbf{I}(y_t = y)$ and $f(y_t,y_{t-1}) = \mathbf{I}(y_t = y)\mathbf{I}(y_{t-1} = y')$.

Then, the distribution of Y given X can be written as

$$p(Y|X) = \frac{1}{Z(X)}\prod_{t=1}^{T}\Psi_t(y_t,y_{t-1},\mathbf{x}_t) \tag{10}$$

where

$$\Psi_t(y_t,y_{t-1},\mathbf{x}_t) = \exp\left\{\sum_{k=1}^{K}\theta_k f_k(y_t,y_{t-1},\mathbf{x}_t)\right\} \tag{11}$$

is the log-linear combination of the feature space. θ_k is the corresponding weight parameter of k-th feature $f_k(\cdot)$, which can be learned by regularized maximum-likelihood estimation (MLE). $K = |\mathscr{F}|$ is the size of feature set.

Appendix C

LSTMs introduce cell state $\mathbf{c}_t$ to cover all information over time. At every time step, $\mathbf{c}_t$ is updated with $\mathbf{c}_{t-1}$ and $\mathbf{z}_t$, which is exactly the same as in Eq. (1). There are two gates, $\mathbf{i}_t$ and $\mathbf{o}_t$, which are calculated by the weight combination of current embedding $\mathbf{x}_t$, last $\mathbf{h}_{t-1}$ and last $\mathbf{c}_{t-1}$, and output through a sigmoid function. This guarantees that each element of $\mathbf{i}_t$,$\mathbf{1}-\mathbf{i}_t$, and $\mathbf{o}_t$ are numbers in [0,1]. After pointwise multiplication operation with another vector with the same shape, they determine the contribution of this vector to the result. In LSTMs, $\mathbf{i}_t$ modulates the combination of $\mathbf{c}_{t-1}$ and $\mathbf{z}_t$, whereas $\mathbf{o}_t$ modulates the contribution of $\mathbf{c}_t$ to $\mathbf{h}_t$:

$$\mathbf{z}_t = \tanh(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1} + \mathbf{b}_z) \tag{12}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{V}_i\mathbf{c}_{t-1} + \mathbf{b}_i) \tag{13}$$

$$\mathbf{c}_t = (1-\mathbf{i}_t)\mathbf{c}_{t-1} + \mathbf{i}_t\mathbf{z}_t \tag{14}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{V}_o\mathbf{c}_{t-1} + \mathbf{b}_o) \tag{15}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \tag{16}$$

$\mathbf{W} \in \mathbb{R}^{d_h \times d_e}$, $\mathbf{U} \in \mathbb{R}^{d_h \times d_h}$, $\mathbf{V} \in \mathbb{R}^{d_h \times d_h}$, and $\mathbf{b} \in \mathbb{R}^{d_h}$ are weight and bias parameters to be learned. These parameters are time invariant, and reduce the size of the hypothesis space.

Appendix D

We present the formula used to calculate $\log(\sum_i e^{s(X,y^i)})$ in Eq. (5). This term is a form of log-sum-exp, and we rewrite it as $\underset{\forall y_{1:T}}{\text{LSE}}\, s(X_{1:T}, y_{1:T})$. Like the forward algorithm in CRF, a middle variable $\delta_t(k)$ is introduced:

$$\begin{aligned}
\delta_t(k) &\triangleq \underset{\forall y_{1:t} \cap y_t = k}{\text{LSE}}\, s(X_{1:t}, y_{1:t}) \\
&= \underset{\forall i, y_{1:t} \cap y_t = k}{\text{LSE}}\, (s(X_{1:t}, y_{1:t-1})|_{y_{t-1}=i} + \mathbf{M}_{ik} + P_{t,k}) \\
&= \underset{\forall i, y_{1:t} \cap y_t = k}{\text{LSE}}\, (s(X_{1:t}, y_{1:t-1})|_{y_{t-1}=i} + \mathbf{M}_{ik}) + P_{t,k} \\
&= \underset{\forall i, y_t = k}{\text{LSE}}\, (\underset{\forall y_{1:t-1} \cap y_{t-1} = i}{\text{LSE}}\, [s(X_{1:t-1}, y_{1:t-1}) + \mathbf{M}_{ik}]) + P_{t,k} \\
&= \underset{\forall i, y_t = k}{\text{LSE}}\, (\underset{\forall y_{1:t-1} \cap y_{t-1} = i}{\text{LSE}}\, s(X_{1:t-1}, y_{1:t-1}) + \mathbf{M}_{ik}) + P_{t,k} \\
&= \underset{\forall i, y_t = k}{\text{LSE}}\, (\delta_{t-1}(i) + \mathbf{M}_{ik}) + P_{t,k}
\end{aligned}$$

Finally, we obtain $\log(\sum_i e^{s(X,y^i)}) = \underset{\forall y_{1:T}}{\text{LSE}}\, s(X_{1:T}, y_{1:T}) = \sum_k \delta_T(k)$.

References

[1] T.D. Gunter, N.P. Terry, The emergence of national electronic health record architectures in the United States and Australia: models, costs, and questions, J. Med. Internet Res. 7 (1) (2005) e3.

[2] Özlem Uzuner, Yuan Luo, Peter Szolovits, Evaluating the state-of-the-art in automatic de-identification, J. Am. Med. Inform. Assoc. 14 (5) (2007) 550–563.

[3] Amber Stubbs, Christopher Kotfila, Özlem Uzuner, Automated systems for the de-identification of longitudinal clinical narratives: overview of 2014 i2b2/UTHealth shared task Track 1, J. Biomed. Inform. 58 (2015) S11–S19.

[4] Amber Stubbs, Michele Filannino, Özlem Uzuner, De-identification of psychiatric intake records: overview of 2016 CEGS N-GRID shared tasks Track 1, J. Biomed. Inform. (2017).

[5] Stéphane Meystre, Shuying Shen, Deborah Hofmann, Adi Gundlapalli, Can physicians recognize their own patients in de-identified notes? in: Studies in Health Technology and Informatics, vol. 205, 2014, pp. 778–782.

[6] Cyril Grouin, Rue John von Neuman, Nicolas Griffon, Aurélie Névéol, Is it possible to recover personal health information from an automatically de-identified corpus of French EHRs? in: Sixth International Workshop On Health Text Mining And Information Analysis (Louhi), 2015, p. 31.

[7] Ralph Grishman, Beth Sundheim, Message understanding conference-6: a brief history, in: COLING, vol. 96, 1996, pp. 466–471.

[8] GuoDong Zhou, Jian Su, Named entity recognition using an HMM-based chunk tagger, Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2002, pp. 473–480.

[9] Andrew McCallum, Wei Li, Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons, Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, vol. 4, Association for Computational Linguistics, 2003, pp. 188–191.

[10] Hideki Isozaki, Hideto Kazawa, Efficient support vector classifiers for named entity recognition, Proceedings of the 19th International Conference on Computational Linguistics, vol. 1, Association for Computational Linguistics, 2002, pp. 1–7.

[11] Thomas G. Dietterich, Machine learning for sequential data: a review, Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), Springer, 2002, pp. 15–30.

[12] Hui Yang, Jonathan M. Garibaldi, Automatic detection of protected health information from clinic narratives, J. Biomed. Inform. 58 (2015) S30–S38.

[13] Yoshua Bengio, Aaron Courville, Pascal Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.

[14] Yann LeCun, Yoshua Bengio, Geoffrey Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[15] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, Pavel Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. 12 (Aug) (2011) 2493–2537.

[16] Sepp Hochreiter, Jürgen Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[17] Zhiheng Huang, Wei Xu, Kai Yu, Bidirectional LSTM-CRF Models for Sequence Tagging, 2015. Available from: arXiv preprint < 1508.01991 > .

[18] Jason P.C. Chiu, Eric Nichols, Named entity recognition with bidirectional LSTM-CNNs, Trans. Assoc. Comput. Linguist. 4 (2016) 357–370.

[19] Xuezhe Ma, Eduard Hovy, End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF, 2016. Available from: arXiv preprint < 1603.01354 > .

[20] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, Chris Dyer, Neural Architectures for Named Entity Recognition, 2016. Available from: arXiv preprint < 1603.01360 > .

[21] Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, Peter Szolovits, De-identification of Patient Notes with Recurrent Neural Networks, 2016. Available from: arXiv preprint < 1606.03475 > .

[22] Laura Chiticariu, Yunyao Li, Frederick R. Reiss, Rule-based information extraction is dead! Long live rule-based information extraction systems!, in: EMNLP, October 2013, pp. 827–832.

[23] Jonathon Read, Rebecca Dridan, Stephan Oepen, Lars Jørgen Solberg, Sentence boundary detection: a long solved problem? in: Coling 2012, December 2012, pp. 985–994.

[24] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, Efficient Estimation of Word Representations in Vector Space, 2013. Available from: arXiv preprint < 1301.3781 > .

[25] Razvan Pascanu, Tomas Mikolov, Yoshua Bengio, On the difficulty of training recurrent neural networks, ICML (3 2) (2013) 1310–1318.

[26] Yoshua Bengio, Patrice Simard, Paolo Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE Trans. Neural Networks 5 (2) (1994) 157–166.

[27] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutn\'\ik, Bas R Steunebrink, Jürgen Schmidhuber, LSTM: A Search Space Odyssey, 2015. Available from: arXiv preprint < 1503.04069 > .

[28] Xiang Zhang, Junbo Zhao, Yann LeCun, Character-level convolutional networks for text classification, in: Advances in Neural Information Processing Systems, 2015, pp. 649–657.

[29] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[30] Nancy Chinchor, The statistical significance of the MUC-4 results, in: Proceedings of the 4th Conference on Message Understanding, 1992, pp. 30–50.

[31] E.W. Noreen, Computer-intensive Methods for Testing Hypotheses: An Introduction, 1989.

[32] Bin He, Yi Guan, Jianyi Cheng, Keting Cen, Wenlan Hua, CRFs based de-identification of medical records, J. Biomed. Inform. 58 (2015) S39–S46.

[33] B. Wellner, M. Huyck, S. Mardis, J. Aberdeen, A. Morgan, L. Peshkin, A. Yeh, J. Hitzeman, L. Hirschman, Rapidly retargetable approaches to de-identification in medical records, J. Am. Med. Inform. Assoc. 14 (5) (2006) 564–573.

Publication 4

A study of EMR-based medical knowledge network and its applications

Chao Zhao, Jingchi Jiang, Zhiming Xu, and Yi Guan

Computer Methods and Programs in Biomedicine 143 (2017) 13–23

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

A study of EMR-based medical knowledge network and its applications

Chao Zhao, Jingchi Jiang, Zhiming Xu, Yi Guan*

School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China

ARTICLE INFO

Article history:
Received 19 April 2016
Revised 23 January 2017
Accepted 9 February 2017

Keywords:
Electronic medical record
Medical knowledge network
Complex network
Knowledge representation

ABSTRACT

Background and Objective: Electronic medical records (EMRs) contain an amount of medical knowledge which can be used for clinical decision support. We attempt to integrate this medical knowledge into a complex network, and then implement a diagnosis model based on this network.

Methods: The dataset of our study contains 992 records which are uniformly sampled from different departments of the hospital. In order to integrate the knowledge of these records, an EMR-based medical knowledge network (EMKN) is constructed. This network takes medical entities as nodes, and co-occurrence relationships between the two entities as edges. Selected properties of this network are analyzed. To make use of this network, a basic diagnosis model is implemented. Seven hundred records are randomly selected to re-construct the network, and the remaining 292 records are used as test records. The vector space model is applied to illustrate the relationships between diseases and symptoms. Because there may exist more than one actual disease in a record, the recall rate of the first ten results, and the average precision are adopted as evaluation measures.

Results: Compared with a random network of the same size, this network has a similar average length but a much higher clustering coefficient. Additionally, it can be observed that there are direct correlations between the community structure and the real department classes in the hospital. For the diagnosis model, the vector space model using disease as a base obtains the best result. At least one accurate disease can be obtained in 73.27% of the records in the first ten results.

Conclusion: We constructed an EMR-based medical knowledge network by extracting the medical entities. This network has the small-world and scale-free properties. Moreover, the community structure showed that entities in the same department have a tendency to be self-aggregated. Based on this network, a diagnosis model was proposed. This model uses only the symptoms as inputs and is not restricted to a specific disease. The experiments conducted demonstrated that EMKN is a simple and universal technique to integrate different medical knowledge from EMRs, and can be used for clinical decision support.

1. Introduction

The electronic medical record (EMR) is the storage of all health care data and medical history of a patient in an electronic format [17]. These data include abundant medical knowledge, such as the current clinical diagnosis, medical history, results of investigations, treatment plans, and so on [12]. This is a novel and rich resource for clinical research [26,44]. As the quantity of EMRs increase rapidly, medical professionals are overwhelmed by this ever-expanding knowledge. Therefore, several effective methods are being developed to assist these professionals [47].

Entities, and relationships between entities, are the primary carriers of medical knowledge in EMRs, and can be extracted by a natural language processing (NLP) technique [44]. Thus, entity recognition [25] and entity relationship extraction become the key tasks in the knowledge extraction of EMRs [29,47]. Additionally, these tasks are an application for NLP in biomedical informatics [38], and have become the basis of many other tasks [28,37].

If we regard the extracted entities as nodes, and entity relationships between two entities as edges, the knowledge derived from EMRs can be represented as a network. Network based methods have been extensively used for medical knowledge representation and inference. These networks can be divided into two major

* corresponding author.
E-mail addresses: zhaochaocs@gmail.com (C. Zhao), jiangjingchi0118@163.com (J. Jiang), xuzm@hit.edu.cn (Z. Xu), guanyi@hit.edu.cn (Y. Guan).

http://dx.doi.org/10.1016/j.cmpb.2017.02.016
0169-2607/

groups: Bayesian networks (probabilistic graphical models, more universal), and complex networks.

Many existing studies have attempted to diagnose or predict diseases based on Bayesian networks [9,13,14,19,27,42]. One of the earliest, and most renowned diagnosis system is *Pathfinder*, which is a decision-theoretic expert system for hematopathology diagnosis [19]. Since that time, the Bayesian networks have played important roles in successive studies. Klann et al. [27] implemented an adaptive recommendation system to recommend a next order of treatment menu, based on the previous orders. Velikova et al. [42] built a probabilistic disease model for preeclampsia using the temporal Bayesian network and implemented it as part of a real-world home-monitoring system for personalized pregnancy care. Flores et al. [13] presented a methodology for incorporating expert knowledge as structural priors for learning the Bayesian networks, and applied it to the study of heart failures. As a probabilistic inference paradigm, Bayesian networks are suitable to model complex interactions between medical entities, and have an even higher accuracy than clinicians in some cases. However, the computational complexity of the structure, and parameter learning of Bayesian networks can become intractable once the number of nodes becomes larger [7]. Hundreds of nodes are difficult for Bayesian networks to handle, thus, the network can only be used in a single, specific field.

On the contrary, the complex network based models can analyze the large-scale data appropriately, however, this is not a specialized model for knowledge representation and inference. Complex networks refer to the large scale networks having small-world and scale-free properties [2,45]. Compared with the random networks, in which any two nodes are independently connected with a fixed probability [11,41], complex networks have a small number of nodes with a much larger number of connections. Studies focused on the network system have pointed out that many real networks are complex networks, rather than random networks [8,10].

In the medical domain, the complex network plays an instrumental role by providing conceptual insights, as well as offering visual and computational methodology [15]. It is widely applied to disease-gene [16], protein-protein [43], disease-symptom [51] interaction analysis, as well as the field of pharmacology [49,50], epidemiology [20], genetics [30] and brain science [24]. Goh et al. [16] proposed *diseasome*, a bipartite graph consisting of the disease and gene nodes. A disease and gene was connected, if mutations in that gene were implicated in that disease. Zhou et al. [51] constructed a symptom-based human disease network from biomedical literature databases, and investigated the connection between clinical manifestations of diseases and their underlying molecular interactions. Tachimori et al. [39] constructed a medical network utilizing a medical textbook. This network had the small-world and scale-free features, but did not show a knowledge inferring ability. To our knowledge, medical networks derived from EMRs have not yet been reported.

We believe such complex networks can support clinical decisions. Applying the artificial intelligence (AI) approach to disease diagnostics is receiving significant attention in the field of medical informatics. Typically, existing studies of AI diagnosis focus on one specific disease. Alizadehsani et al. [1] selected four combinations of related features for coronary artery disease. Then, they applied sequential minimal optimization, and other algorithms, in these groups of features and reached the best accuracy of 94.08%. Rau et al. [35] proposed a prediction model for developing liver cancer in type 2 diabetes mellitus patients. They identified 10 risk factors as variables, then constructed an artificial neural network (ANN) and logistic regression (LR) prediction models. The best results of sensitivity and specificity were 0.757 and 0.755, respectively. Hariharana et al. [18] selected 22 raw features from the voice signals of people diagnosed with Parkinson's disease. Their diagnosis model consisted of feature pre-processing, feature reduction/selection, and different feature classifiers, using a support vector machine (SVM) and neural network. They obtained a classification accuracy of 100% for the test set.

In these studies, corresponding features of patients were manually selected, then the diagnostic problems were converted into classification tasks. After the feature selection and reduction was completed, classifiers were used to discriminate whether the patients had the disease. It could be observed that the classifier of a specific disease could obtain sufficient results to be used for clinical experience, and support a clinical decision. However, this type of study focused more on feature extraction and feature pre-processing. Good features can usually get satisfactory performance, while bad features cannot accurately distinguish the positive cases from negative ones. Good feature selection needs professional medical knowledge. Moreover, the model can only be applied to a certain disease. If people have certain symptoms, and want to know what affliction they might have, these systems are not of much help. We attempt to exploit the possibility of constructing a more universal diagnosis model, utilizing simple features (like symptoms) as inputs and not restricting it to a specific disease.

In this paper, we construct **EMKN**, a new **M**edical **K**nowledge **N**etwork based on **E**MRs. Nodes of this network are medical entities and edges are co-occurrence relationships between entities in the same records. The more frequently an entity pair occurs together, the larger the edge weight between this pair becomes. We calculate primary quantities of EMKN and validate its complex properties. Next, we propose a basic, universal diagnosis model using EMKN to show that EMKN can indeed express medical knowledge. We apply this model to real EMRs and prove its effectiveness.

There may be two innovations in this paper:

- We construct EMKN, an EMR-based medical knowledge network, to represent medical knowledge. Then, we find the complex properties of this network.
- We propose a simple, but universal diagnosis model based on the network. This model takes symptom entities as inputs, and can be used for multi-disease diagnosis.

The remainder of this paper is structured as follows. In Section 2, we give a brief introduction to our EMR corpus, then provide the construction approach and property analysis of EMKN. In Section 3, we describe the details of the diagnosis model. Then, we evaluate our model using actual records in Section 4, and give the results with discussion. Conclusion and future studies will be presented in Section 5.

2. Medical knowledge network construction

2.1. Medical entities and assertions

Referencing the *medical concept annotation guideline* and *assertion annotation guideline* given by Informatics for Integrating Biology and the Bedside (i2b2) [21,22], we have created guidelines for Chinese EMRs, and manually annotated 992 records as the corpus under the guidance of medical professionals [46]. These records were retrieved from The Second Affiliated Hospital of Harbin Medical University, and contained 887 individual patients. We have obtained the usage rights for these records. [1] We manually annotated the medical entities and their assertions, which denotes modifiers. In the annotating process, we classified the entities into

[1] https://github.com/WILAB-HIT/Resources.

Table 1
Weight adjustment rules.

Types of the node pair	Edge weight adjustment
Present-present	1.0
Present-absent	−1.0
Other conditions	0.5

five classes, such as *disease, disease type, symptom, test,* and *treatment*. There were seven assertions for disease and symptom entities, namely *present, absent, family, conditional, possible, hypothetical,* and *occasional,* and another three assertions for treatment, such as, *present, absent,* and *history*.

Fig. A.1 in Appendix A shows a de-identified progress note from our records corpus. A standard progress note contains four components: chief complaint, symptoms and test results, assessment and diagnosis, and treatment plan. In addition, we list the corresponding annotations about this progress note in Table A.1.

2.2. Network construction

We used EMKN to integrate the large amounts of medical entities into a complex network. In this network, the nodes corresponded to entities, and edges to the co-occurrence relationships between an entity pair in the same record. We should have used the exact entity relationships as edges directly, however, as of yet, our extraction technique of entity relationships has not performed well enough to support such a method. Thus, we weakened the relationship using co-occurrence, and constructed an undirected network. The co-occurrence of medical entities is also be utilized for association analysis of diseases and symptoms [23,34]. We connected a pair of nodes using an edge, if and only if:

- the two nodes occurred together in a record; and
- there was, at minimum, one node with the annotation as present.

The weight of the edge was adjusted according to the rules in Table 1 :

The motivation of these rules lies within the following aspects: an entity pair with no present assertion tells us little useful knowledge, thus, we ignore such pairs. A pair with both present assertions indicates a certain positive correlation between them, thus, we increase the weight by 1. If there is only one present assertion, and the other assertion is neither present nor absent, the confidence is decreased. In this case, for example, we roughly discounted the weight increase to 0.5 for present-conditional pairs. On the contrary, the present-absent pair shows a negative correlation, and the weight is decreased by 1.

As new records were added in EMKN, we dynamically increased or decreased the weight of the edge according to the rules. After adding all 992 records in the graph and filtering the edges with a negative weight, we obtained a medical knowledge network with 6733 nodes and 154,462 edges.

As mentioned above, we would use a diagnosis task to confirm the effectiveness of EMKN. Therefore, we focus more on the relationships between disease and symptom entities. We reserve these two kinds of entities and the edges among these entities, then, we obtain a new Symptom-Disease EMKN (SDEMKN), as shown in Fig. 1 . This network possesses 4675 nodes and 19,564 edges.

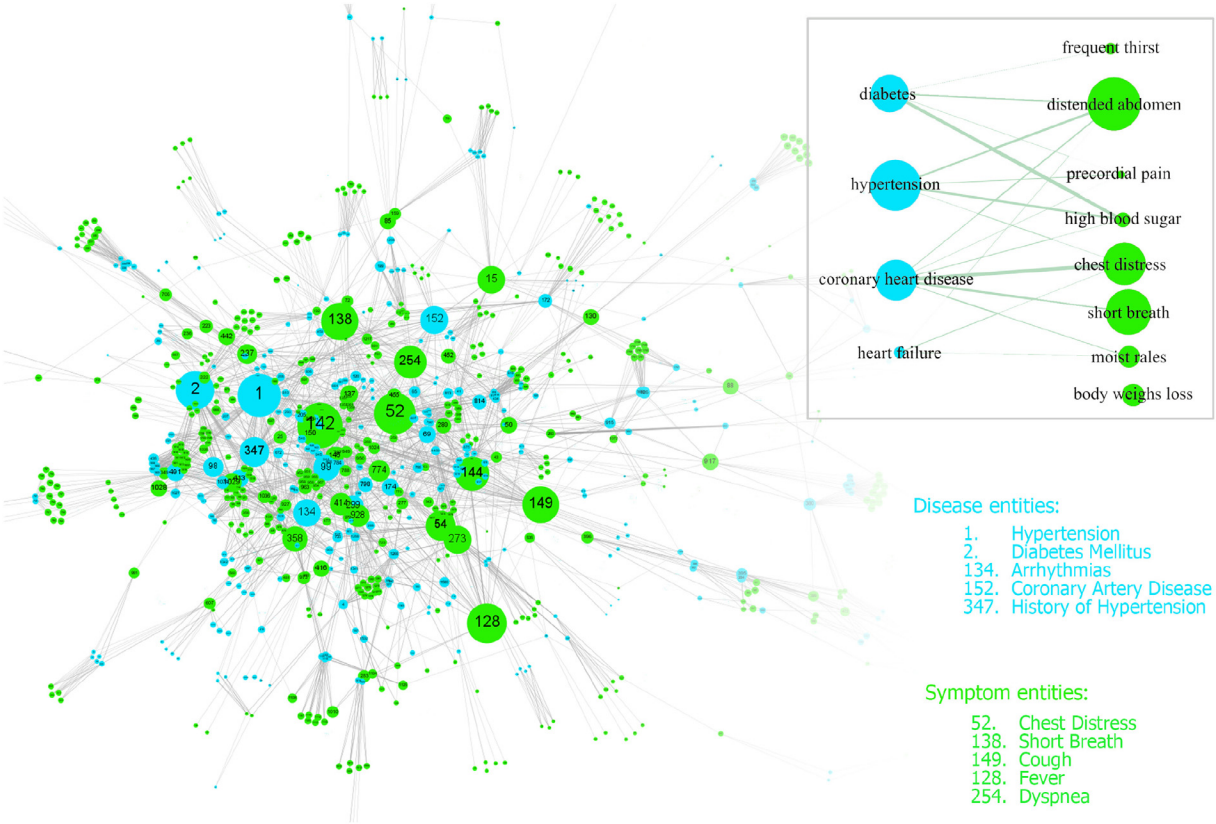


Fig. 1. The Symptom-Disease EMKN visualized by Gephi [3]. The main figure depicts a miniature version of the symptom-disease EMKN, which is generated by 150 records. It is a bipartite network containing only symptom and disease entities. We represent the disease entities using blue nodes and symptom entities using green nodes. The size of the nodes indicates their degree, and the thickness of edges indicates their weight. The label of each node is its unique ID. The top right corner shows a tiny subgraph of SDEMKN. The bottom right corner represents five entities with large degree for each type. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

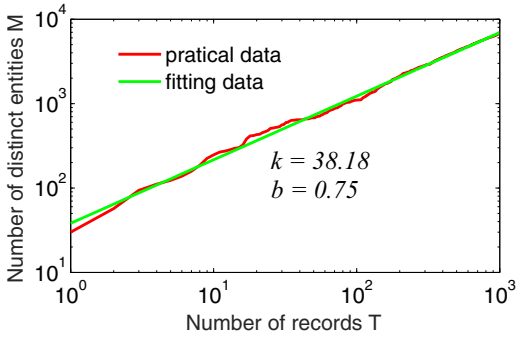


Fig. 2. Curves of the node increasing rate. This figure shows the fit of the actual data using Heaps law. The red curve is practical data and the green line is the fitting result. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2.3. Network property analysis

To explore the EMKN further, first, we analyze several quantities of this network, such as the growth rate of nodes, the average length, the clustering coefficient, and so on.

2.3.1. Network quantities

As the number of records increase in EMKN, the increasing tendency of nodes follows the Heaps Law, as shown in Fig. 2 . Heaps Law is used to estimate the vocabulary size in the field of Information Retrieval (IR) and is in the following form:

$$M = kT^b$$

Here M is the number of distinct entities and T is the number of records. k and b are two parameters to be estimated. The relationship between T and M is linear in a log-log space. Using this line to fit our data, we get $k = 38.18$ and $b = 0.75$.

With the improvement of entity auto-extraction technology, it is not difficult to add tens of thousands of records to the network. Heaps law tells us that the growth rate of entities would eventually tend to be slower than the linear function. If we mapped the entities with the same semantic meaning in a unified node, the growth rate might be slowed even further. It guarantees that the calculation based on this network would not be too large. Meanwhile, a network constructed by large amounts of records can cover most of the entities. When we take a new record to infer the diseases, the possibility to meet the unlisted entities can be reduced greatly. We will observe the importance of this in Section 4.

The following gives the definition of certain basic quantities and properties in the field of complex networks [32].

The average path length L is defined as the mean shortest distance between two nodes in the network. If we select two nodes i and j, we can get the shortest distance d_{ij} between them as long as there is at least one connecting path. A connected, undirected graph with N nodes has $\frac{1}{2}N(N+1)$ distinct node pairs, thus, the average path length of this graph can be calculated as

$$L = \frac{1}{\frac{1}{2}N(N+1)} \sum_{i \geq j} d_{ij}.$$

If there is more than one connected component in the graph, we simply ignore the pairs with no connecting paths.

For a certain node i with a degree of k_i, it has k_i neighbors. These k_i nodes can have at most $k_i(k_i-1)/2$ edges. However, there are exactly E_i edges among these neighbors. The coefficient factor of this node i is defined as the ratio of these two quantities

$$c_i = \frac{2E_i}{k_i(k_i-1)}.$$

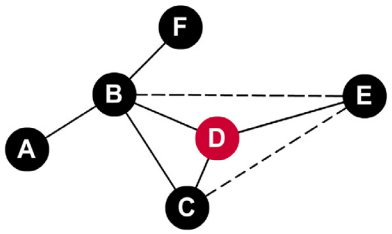


Fig. 3. An example to calculate the coefficient factor of node D. There are three neighbors of node D, thus $K_D = 3$. Among these neighbors, there can be $K_D(K_D-1)/2 = 3$ edges, but only the edge between B and C exists (the dashed lines do not really exist), thus $E_D = 1$. The coefficient factor of node D can be calculated by $c_D = 2E_D/[K_D(K_D-1)] = 1/3$.

Fig. 3 shows an example to calculate c_i. Similarly, if k_i is less than 2, we simply ignore this node by setting c_i as 0.

The perception behind the formula is as follows: if the node j and node k are both connected with node i, there is a heightened possibility that j is also connected with k, just like a friend of your friend is more likely to be a friend of yours. The coefficient factor is designed to quantify the possibility. From a different perspective, the coefficient factor reflects the "density" of the triangles among the node i and its neighbors. Every node in a complete graph has a coefficient factor of 1. From this perspective, the definition of the clustering coefficient of the whole network is easier to understand. It is defined as the mean coefficient factor in the network:

$$C = \frac{1}{n} \sum_{i=1}^{n} c_i.$$

Table 2 reflects the above quantities of EMKN and SDEMKN compared with a random network of a similar size.

2.3.2. The small-world and scale-free properties

Complex networks have small-world and scale-free properties [32]. The small-world property tells us that most pairs of nodes have short paths throughout the network. This indicates that even if we have millions of nodes, we can usually move from one arbitrary node to another in several steps. The "small-world" comes from the famous small-world experiment, which found that two strangers can be connected in an average of six steps [31]. More accurately, compared with a random network of the same size, the complex network has a similar average length L but a much higher clustering coefficient. That is

$$\gamma = \frac{C_{comp}}{C_{rand}} \gg 1$$

$$\lambda = \frac{L_{comp}}{L_{rand}} \sim 1,$$

where the subscript *comp* denotes the complex network, and the subscript *rand* denotes the random network.

The scale-free property indicates that the degree distribution of the nodes follows a power law

$$P(k) \propto k^{-\alpha},$$

where $P(k)$ is the probability that a randomly selected node has the degree k. In other words, the proportion of nodes with degree k is closer to $k^{-\alpha}$. This is a very different property compared with the random network, where the degree distribution is binomial or Poisson. The latter distribution has a peak at the degree k, and the possibility that the degree of a node is considerably larger than $< k >$ decreases rapidly. However, it is difficult to estimate the largest degree of the nodes in the power law distribution because it has a long right tail of values that are far above the mean.

If we plot $P(k)$ in a log-log space, it would be approximately linear, because $\log P(k) \propto -\alpha \log k$. However, when the degree is very

Table 2
The quantities of EMKN, SDEMKN and a random network (RN) with similar size computed by Gephi.

Network type	Nodes	Edges	Average degree	Average path length	Diameter	C
EMKN	6733	154,462	45.88	2.26	4	0.839
SDEMKN	4675	19,564	8.37	4.15	10	0.797
RN	6733	158,669	23.566	2.71	4	0.007

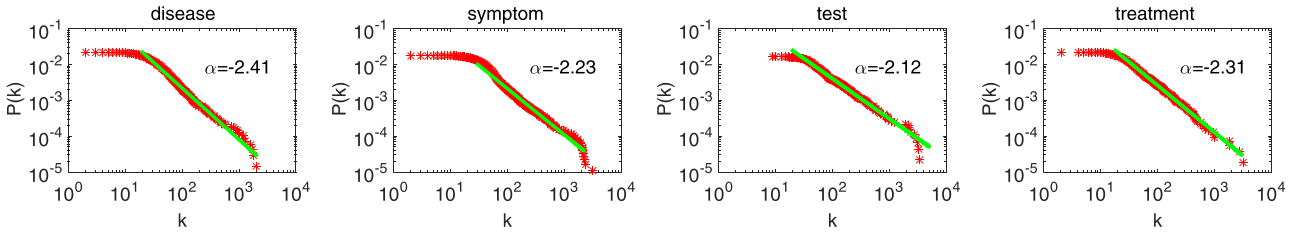


Fig. 4. Degree distribution of four classes in EMKN.

large, there are only 1, 2, or 0 nodes to be the numerator in which to calculate $P(k)$. These discrete and oscillatory numerators make the tail of the distribution very noisy. To avoid this, we use cumulative distribution instead. The cumulative distribution also follows the power law [33]

$$P_c(k) \propto \sum_{k'=k}^{\infty} k'^{-\alpha} \propto k^{-(\alpha-1)},$$

but with the exponent being $\alpha - 1$ rather than α. This can be easily derived as soon as we shift to a continuous perspective, such that $\int_x^{\infty} x'^{-\alpha} dx' \propto x^{-(\alpha-1)}$.

The contrast of data from Table 2 shows that the average path length of EMKN is similar with the length of the random network, but the clustering coefficient C is much higher. This phenomenon reflects the small-world properties of EMKN. SDEMKN has similar properties. To validate the scale-free properties of the EMKN, we plot the cumulative distribution for each class of nodes, except the class of disease types. This is because the number of this class is much smaller than others. As Fig. 4 shows, we can observe that most distributions follow a truncated power law well. The symptom class and disease class in SDEMKN have similar properties.

The scale-free property shows that, EMKN has tiny amounts of core nodes with a much higher degree. These nodes connect the whole network together. From a diagnosis view, several diseases, such as hypertension and diabetes, appear with many symptoms, and it might be easier to associate them with secondary diseases. On the other hand, some symptoms, like dizziness and chest distress, can be viewed as a warning of disease, however, there is a relatively weak guiding significance for the diagnosis because there are too many disease neighbors.

2.3.3. Network community structure

The 992 records in the dataset are from 36 different departments of the hospital. Based on the above work, we integrated all the entities of these records into one network. However, the symptoms and diseases in one department are quite different from other departments. We want to exploit whether the network has a certain community structure that is related to the department information.

A network with a community structure indicates that the network has some groups of nodes that have a high density of edges within them and a lower density of edges between the groups. We ran the structure extracting algorithms [5] in the SDEMKN and obtained 649 groups, or modularity classes. Next, we selected the 12 largest classes to show their connections with the real department. The quantity of entities in the 12 classes account for 58.28% of all entities in the network. Fig. 5 shows the department distributions of the 12 modularity classes, respectively. Table 3 shows more details of Fig. 5. For each modularity, we show the number of the entities that are in the modularity, coupled with the information of its typical departments, including the ID numbers and department names. We also calculate the percentage of entities in each modularity class that falls in its typical departments. The typical departments consist of the department having the most entities in such class, as well as the departments with a number more than one third of the number of the largest department in such class.

From Table 3 and Fig. 5, we can see that there are obvious and direct correlations between the community structure and the real department classes in the hospital. For some specific modularities (like modularity 173, 247, and 263), more than 95% of the entities are from the same department. The community structure of MKN can avoid the global search of the network during the inference process. For instance, in diagnosis inference, when most of the symptoms come from the same department, we can only consider the diseases in the communities corresponding to that department.

3. Disease inference

In medical diagnosis, we are typically faced with a set of symptoms and test results. Then, we attempt to find the cause behind them. To make use of the medical knowledge of EMKN, we designed a diagnosis model, which can infer the corresponding diseases with the given symptoms.

We think symptoms are caused by diseases. If we view the symptoms as vectors based on the diseases, we can describe the causality to some degree. Suppose there are N distinct disease entities. Then, the symptom vector also has N dimensions. Further, we can represent symptom set S as the sum of all the symptom elements. Next, we evaluate the cosine similarity between a certain disease and the symptoms set, and find diseases with higher similarity to the diagnosis results. We suppose that these diseases would have a greater possibility to be the actual diagnosis results, or they would have certain reference value for medical professionals. The generation of vectors is described in detail below.

A certain symptom s_i can be represented as a vector in the space spanned by diseases like follows:

$$\mathbf{V}(s_i) = [R(s_i, d_1), R(s_i, d_2), \cdots, R(s_i, d_N)]^T,$$

where $R(s_i, d_j)$ represents the relevance between s_i and the $j-th$ disease d_j.

Referencing the idea of TF-IDF [48], and the analysis of the network properties in Section 2.3, we had a perception that the higher edge weight between a disease and symptom pair depicted a stronger relevance between them. On the contrary, a symptom with a higher degree has less guiding significance for diagnosis. For example, “diabetes” has an edge with “excessive thirst”

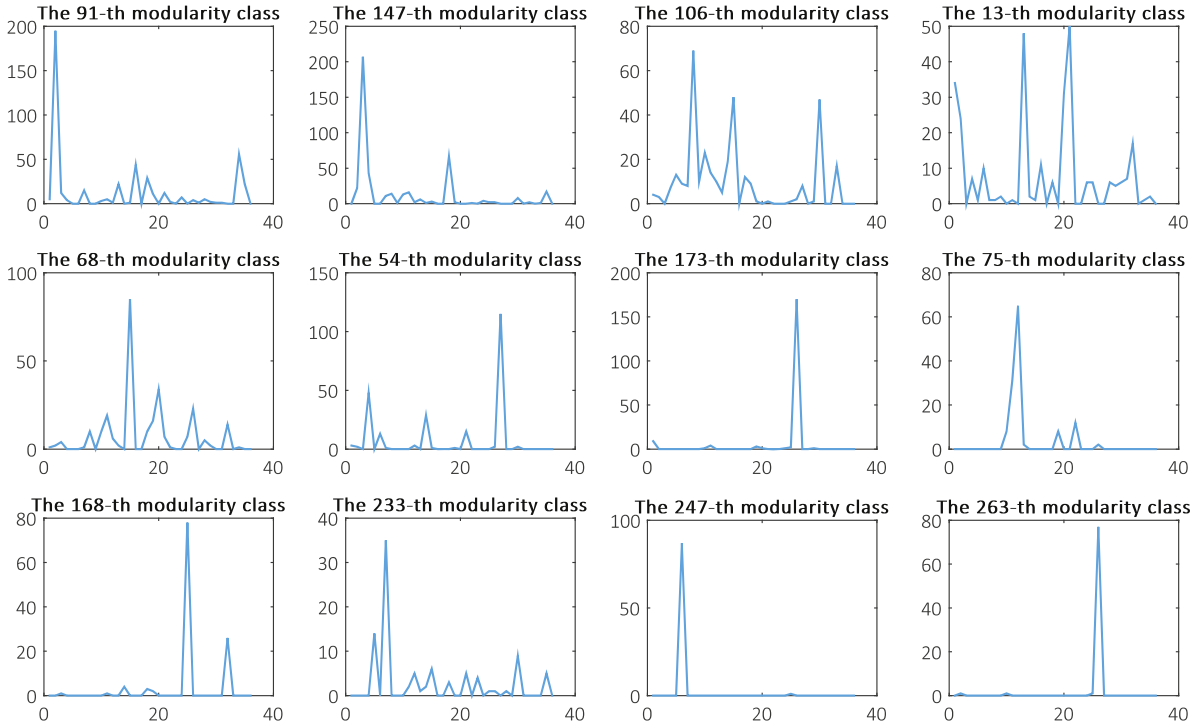


Fig. 5. The corresponding relationships between modularity classes and departments of hospitals. The x-axis indicates the 36 departments and the y-axis represents the number of entities in a certain department. The title of the subfigure displays the ID of the modularity class from which the number is counted.

Table 3
This table shows the largest 12 modularity classes of MKN and their corresponding department names. It also lists the percentage of entities in each modularity class that falls in its typical departments in the fourth column.

Modularity class	Number of entities	Department ID	Percentage (%)	Department name
91	460	2	42.39	Department of Cardiology
147	445	3	46.52	Department of Neurology
		8	20.23	Department of Endocrinology
106	341	15	14.08	Department of Hemopathology
		30	13.78	Department of Urology
		21	17.48	Department of Thoracic Surgery
13	286	13	16.78	Department of Respiration Medicine
		1	11.89	Department of General Surgery
68	260	15	32.69	Department of Hemopathology
		20	13.08	Department of Oncological Radiotherapy
54	235	27	48.94	Department of Emergency Surgery
		4	20.43	Department of Neurosurgery
173	192	26	88.54	Department of Obstetrics and Gynecology
75	128	12	50.78	Department of Infectious Diseases
168	115	25	67.83	Department of Paediatric Internal Medicine
233	94	7	37.23	Department of Rheumatism
		5	14.89	Department of Dermatology and Venerology
247	88	6	98.86	Department of Ophthalmology
263	80	26	96.25	Department of Obstetrics and Gynecology

and the weight is 9.5, however, the weight of another edge with "headache" is only 0.5. Thus, we hoped the relevance of the former pair was larger. As another example, both "dizziness" and "numbness on left side" have edges with "cerebral infarction", and the weights of the two edges are both 3.5. However, the degree of "numbness on left side" is only 11, which is much smaller than the former symptom of 129. Thus, again we hoped that the relevance between "numbness on left side" and "cerebral infarction" became larger. Based on the above analysis, we designed the $R(s_i, d_j)$ as follows:

For a given symptom s, we separated the disease nodes in SDEMKN as two sets:

$$\mathcal{A}(d) = \{d| \ < d, s > \text{exist}\}$$
$$\mathcal{B}(d) = \{d| \ < d, s > \text{does not exist}\}$$

If we use $\deg(s_i)$ to indicate the degree of symptom s_i and weight$< d_j, s_i >$ to indicate the edge weight between s_i and d_j, then

$$R(s_i, d_j) = \begin{cases} \text{weight} < \text{s}_\text{i}, \text{d}_\text{j} > \times \log_2 \frac{\text{N}}{\deg(\text{s}_\text{i})} & \text{if } d_j \in \mathcal{A}(d) \\ 0 & \text{if } d_j \in \mathcal{B}(d) \end{cases}$$

After all the disease entities had been considered, we obtained the symptom vector $\mathbf{V}(\mathbf{s_i})$. The disease d_j can also be represented as the same dimensional vector, with the $j-th$ entry in the vector as 1 and others as 0.

For any disease d_i and symptom set $S = \{s_1, s_2, \cdots, s_m\}$, we could calculate the similarity between the symptom vector and a certain disease vector using the cosine similarity directly:

$$\text{sim}(d_i, S) = \frac{\mathbf{V}(d_i) \cdot \mathbf{V}(S)}{|\mathbf{V}(d_i)| \times |\mathbf{V}(S)|},$$

where $\mathbf{V}(S) = \sum_m \mathbf{V}(s_i)$. Because $\mathbf{V}(d_i)$ is a base vector, $\text{sim}(d_j, S) = \mathbf{V}_j(S)/|\mathbf{V}(S)|$.

Next, we ranked the disease list in a descending order according to the similarity score, and returned the probable diagnosis result list.

Naturally, we could consider another vector representation approach, which adopts symptoms as a vector base. We will conduct experiments to compare these two different representations in the next section. Additionally, we will implement another two baseline methods, using the latent semantic indexing (LSI) [36] and logistic regression (LR) model. The four methods are listed as follows:

- disease-base: adopt diseases as a vector base to represent entities
- symptom-base: adopt symptoms as a vector base to represent entities
- LSI: Considering the symptoms and disease vectors are high-dimensional and sparse, we apply LSI to decompose the vectors to 300 dimensions, then use the results of the singular value decomposition (SVD) to calculate the similarity.
- LR: As mentioned earlier, the diagnosis model can be considered as a classification task. For a certain disease, we use the symptom vectors as features and {0, 1} as the output to indicate the results. Thus, we can construct an LR classifier for each disease.

4. Experiment and discussion

4.1. Experiment setup

To evaluate our model, we randomly selected 700 records from our corpus as the training set, and 292 records remained as the test set. We re-constructed the SDEMKN using the new training data. In the process of entity extraction, we added simple rules to unify parts of entities which had the same meanings. For a test record, we took symptoms with positive assertions as input to infer the probable diagnosis result. Diseases in the record with non-absent assertion were regarded as the actual diagnosis result.

We adopted two measures, *Recall*(10) and Average Precision (*AP*), to evaluate the inferred results. *Recall*(10) is the recall (or the true positive rate) of the first 10 results. That is

$$Recall(10) = \frac{\#(\text{true positive diseases returned in top 10 items})}{\#(\text{positive diseases in records})}.$$

It was inspired by *Prec*(10), which is a widely used information retrieval measure that indicates the precision of the first 10 results. If there are four diseases in one record and we return three of them during the first 10 results, then the $Recall(10) = 3/4 = 0.75$, but the $Prec(10) = 3/10 = 0.3$ and the *Prec*(10) can be no longer than 0.4. Thus, we believe *Recall*(10) is a more appropriate measure for current cases. Average precision of the results list is a commonly used measure in IR experiments as well, and has been demonstrated as one of the most stable and discriminate measures [6]. If a record has m actual diagnosis results, and we return n of them, ranked as $r_1, r_2, \cdots, r_n$, respectively, then the *AP* is given by

$$AP = \frac{1}{m}\sum_{i=1}^{n}\frac{i}{r_i}$$

The most ideal condition is that the m results all be returned and ranked at the top of the list, then $AP = 1$.

Although we have more than 4000 medical entities in EMKN, the size of our training set is far from the level that can cover all the entities. There are many unlisted symptoms and diseases in the test records, which can cause underestimation of our model's inferring ability. To avoid this condition, we filtered out the following two kinds of records from the test set:

- Records with less than half the symptoms contained in EMKN (63 records)
- Records with no diagnosis result contained in EMKN (12 records)

Accordingly, the size of the test set declined from 292 to 217 as 25% of the records were discarded. This was a stopgap to overcome the temporary lack of manually annotated corpus. As we mentioned in Section 2.3, if the size of the network is large enough, we have a low possibility to encounter new entities. Additionally, for those records with less than half the diseases contained in EMKN, we can never get a recall of more than 50%. Thus, we only use the diseases contained in EMKN as the standard answer for evaluation.

Moreover, we asked clinicians to manually evaluate 75 records out of the 217, in addition to the automatic evaluation method. We provided the symptom list, and the first 10 diseases obtained by the diagnosis model for each test record. Next, the clinicians were asked to select the correct diseases among them in accordance with their own clinical experiences. The average precision was adopted as the measure to evaluate the results.

4.2. Results

Fig. 6 shows the distribution of *Recall*(10) and *AP* in our filtered test set, while Fig. 7 shows the cumulative distribution of these two measures, as well as the evaluation results made by the clinicians. Diagnosis cases are also shown in Table A.2.

4.3. Discussion

From the contrast of the evaluation measures, we can observe that vector representation using the diseases as a base obtains a better result. 73.3% of the test cases can return at least one actual disease in the first 10 results, and 30.0% can return all of them. This can be interpreted as follows: diseases are the cause of symptoms, and as they have less quantity than symptoms, they are a more appropriate choice for the base vectors. Compared with the symptom-based method, the performance of LSI only increases slightly, since the principle of LSI is co-occurrence. Similarly, our network is based on the relationship co-occurrence of entities. An LR classifier obtains poor results because the positive cases are much fewer than the input dimensions. It is natural for the manual evaluation result to be better than the calculated result, because clinicians can handle the relationship of medical concepts and terms with more flexibility, instead of simply matching terms.

Moreover, we found that 26.7% of records returned with none of the actual diseases in the first 10 results. Such ineffectiveness stems from two points. One is the imperfection of our EMKN. This includes, as we discussed earlier, the small size of the network, the use of co-occurrence relationships rather than exact entity relationships, and so on. On the other hand, when we analyze the records with bad performance, we find that though the model does not return the exact same diseases, it returns many related results. Taking the last case in Table A.2 as an example, when we tested a record with a diagnosis of "space-occupying lesion of the right lung", instead the model returned "space-occupying lesion of the lung" as the first result, because these two entities are both in EMKN. This result still has a significant guide to the diagnosis, though the *Recall*(10) is measured as 0 in our evaluation. This also reveals the difficulty of multi-disease diagnosis using just complaint symptoms as features.

4.4. Limitations of this study

This is a preliminary work for medical knowledge representation and mining, and some problems still exist. First, the dataset we used is manually annotated with the help of medical experts. Although it covers all the departments of the hospital, there are

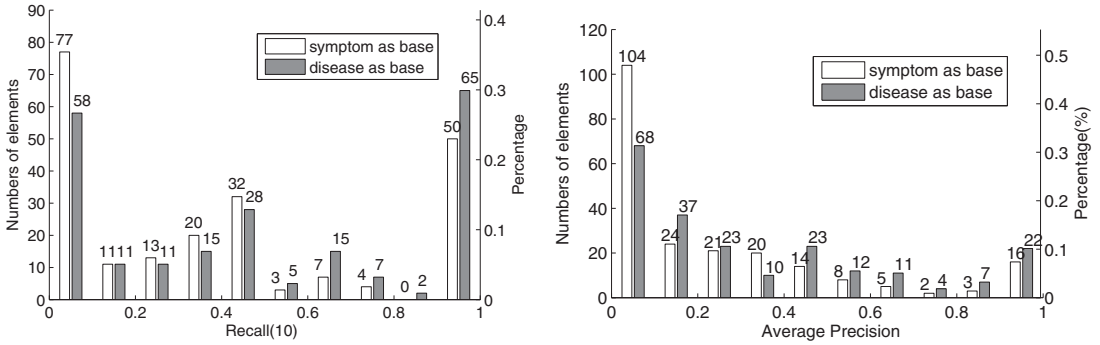


Fig. 6. The distribution of *Recall*(10) and *AP*. The grey bars show the result using the diseases as a base, and the white bar is the symptom-base counterpart. The left y-axis displays the number of records and the right y-axis displays the corresponding percentage.

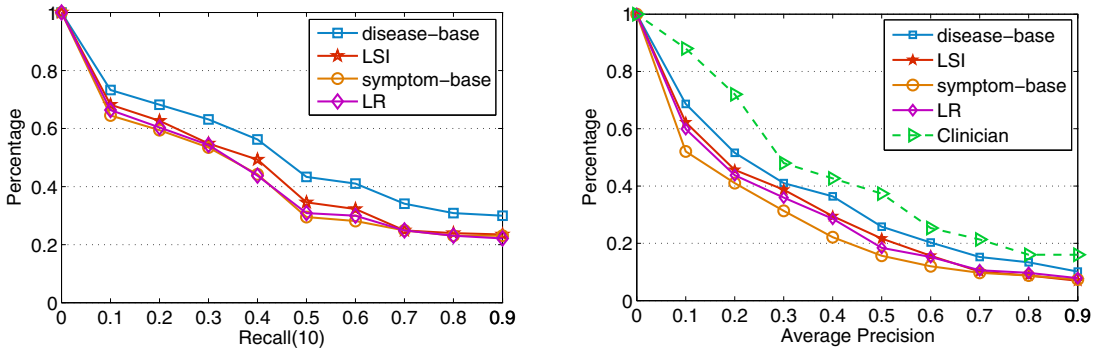


Fig. 7. The cumulative distribution of Recall(10) and *AP*. The test cases with evaluation measures more than the number in the x-axis are counted, and the y-axis displays the corresponding percentage of the count.

only 992 notes, which leads to a limited quantity of records for each department. Based on these records, we have developed an automated system to extract entities and entity relationships via NLP technology. Nevertheless, we still use the annotated records as the dataset to obtain more reliable results. We hope to expand these methods to a more massive record dataset, once the performance of the auto-system can be better.

In addition, the representation of medical entities can be improved further. The representation we used in Section 3 is high-dimensional and sparse. That is to say, the dimension of the vectors equals the amount of the entities and most elements of the vectors are 0. This may lose the relationships between entities. For example, the vector representation of “diabetes” should be more similar with that of “type II diabetes” than that of “pneumonia”, but the current representation cannot depict such a difference. Learning from the distributed representation of words in the field of natural language processing [4,40], we will attempt to represent the medical entities using dense and low-dimensional vectors. The dimension d of these vectors is much lower than the amount of entities (for example, $d = 50$ or $d = 100$), and the elements of vectors are continuous real numbers. Entities with similar semantics would be close in the vector space, which can improve the generalization ability of the models. With the benefit of such generalization, this kind of representation can alleviate the sparsity issues and depict the medical entities with more accuracy.

5. Conclusion

We constructed **EMKN**, an **E**MR-based **M**edical **K**nowledge **N**etwork, using a manually annotated EMR corpus. This network took medical entities as nodes and the co-occurrence relationships of entities in the same record as edges. We obtained main quantities of EMKN and validated its small-world and scale-free properties. In addition, we illustrated that the community structure of this network was associated with the real department information.

According to the subgraph SDEMKN, we represented the entities as vectors based on diseases and proposed a basic diagnosis model. This model took symptoms as inputs and returned the diseases with high similarity to the symptoms as outputs. Empirical results demonstrated the effectiveness of our model. It was further verified that EMKN is a simple and universal technique to integrate different medical knowledge from massive EMRs data, and can be used for medical diagnosis.

Furthermore, we will exploit EMKN on other applications. In the aspect of network construction, if we consider the entities of the test and treatment types during the inference process, EMKN can also support a recommendation system for test and treatment plans. For the entity representation, the vectors of medical entities can be learned further, and used for other tasks such as medical entity similarity calculation and potential entity relationship mining.

Acknowledgment

This work is supported by the Natural Science Foundation of China (No. 61672185). We thank the Second Affiliated Hospital of Harbin Medical University for providing the corpus used in this study. We also thank the anonymous reviewers for their comments, which provided us with significant guidance.

Appendix A

Fig. A.1 and Table A.1 show a progress note from our records corpus, as well as its corresponding annotation samples. Table A.2 shows six real diagnosis cases in Section 4

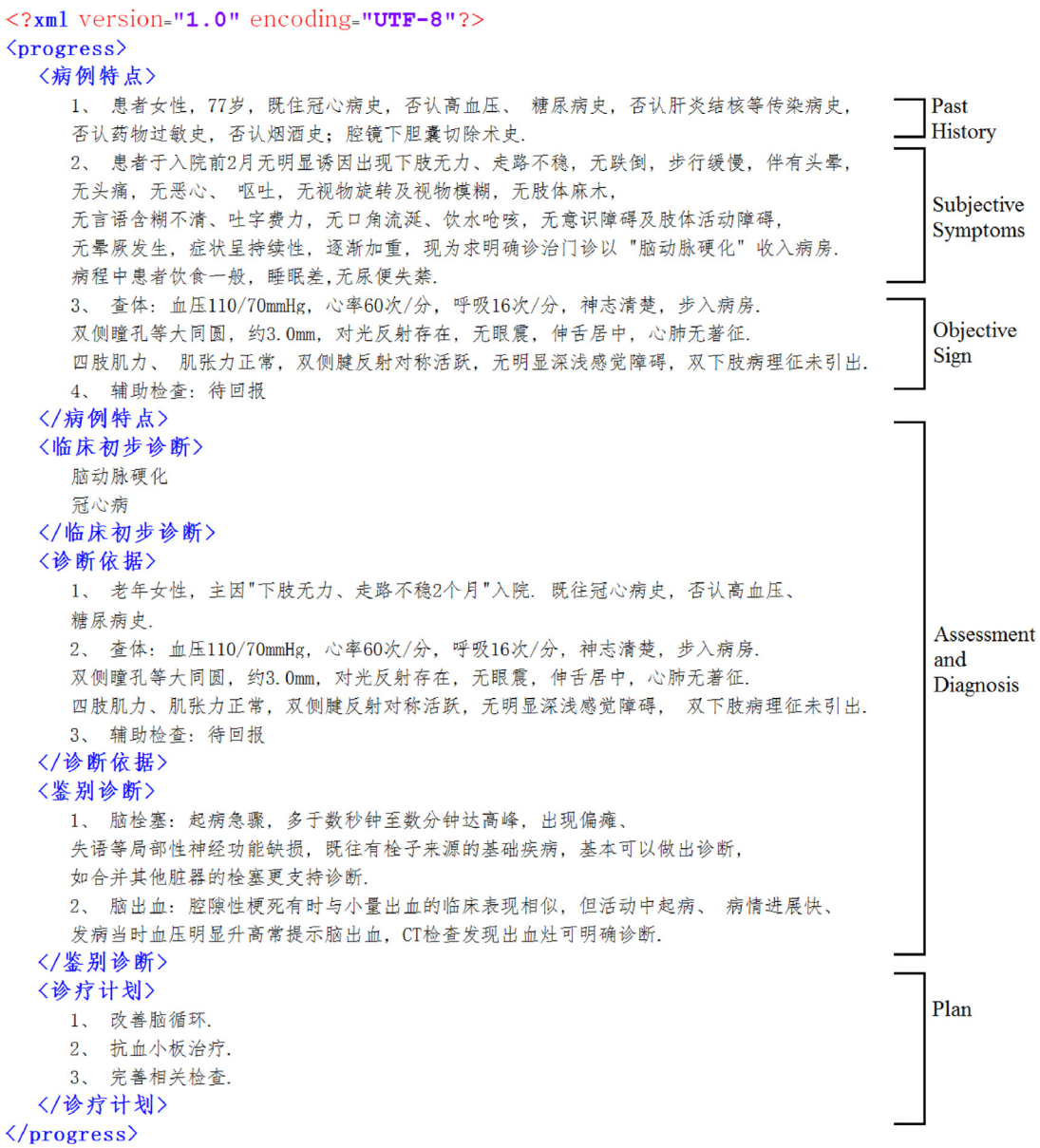


Fig. A.1. A progress note from the records corpus.

Table A.1
Annotations of the entity and assertion information in the record example. We randomly select 14 entities out of 40, because of the space limitations.

Medical entity	Position in the text	Entity type	Assertion type
高血压 hypertension	84:87	disease	absent
糖尿病史 the history of diabates	89:93	disease	absent
腔镜下胆囊切除术史 the history of laparoscopic cholecystectomy	122:131	treatment	history
下肢无力 weakness of bilateral lower limbs	153:157	complaintsymptom	present
走路不稳 staggering when walking	159:163	complaintsymptom	present
头晕 dizziness	178:180	complaintsymptom	present
恶心 nausea	188:190	complaintsymptom	absent
呕吐 vomiting	192:194	complaintsymptom	absent
脑动脉硬化 cerebral arteriosclerosis	290:295	disease	possible
眼震 nystagmus	405:407	testresult	absent
双下肢病理征 pathologic reflexes of bilateral lower limbs	457:463	testresult	absent
冠心病 coronary heart disease	512:515	disease	possible
改善脑循环 improve brain circulation	955:960	treatment	present
抗血小板治疗 antiplatelet therapy	967:973	treatment	present

Table A.2
Some examples of diagnosis cases. We only list the five predicted diseases with the highest possibility, due to the space limitations. If the predicted disease also exists in the actural disease list, we write this disease in bold format.

Symptoms	Actual diseases	Predicted diseases	Recall(10)	Average precision
mental confusion no talking uncooperative in examination pathological sign(+) of the left lower extremity somnolence rave	subarachnoid hemorrhage brain edema temporal bone fracture formaldehyde poisoning sphenoid sinus effusions	**subarachnoid hemorrhage** **brain edema** cerebral infarction cerebral hemorrhage **sphenoid sinus effusions**	1.0	0.796
dizziness headache vomiting slow response to light	head trauma contusion and laceration of the brain skull fracture trauma	**head trauma** **trauma** intracerebral hemorrhage hand injury scalp hematoma	0.75	0.637
polydipsia polyphagia frequent urination irregular pulse rhythm abdominal distention	diabetes stage III hypertension disease arrhythmia PAC cerebral arteriosclerosis	type II diabetes **diabetes** arrhythmia diabetic ketosis dyslipidemia	0.6	0.392
chest pains tightness	coronary heart disease angina pectoris chronic gastritis neuroses	**coronary heart disease** arrhythmia lung space-occupying type II diabetes cardiac insufficiency	0.5	0.293
fever edema swollen lingual papillae pain red and swollen gums dispersed ulcers	post breast cancer infectious fever agranulocytosis chronic viral hepatitis agranulocytosis anaerobic infection EBV infection HCMV infection	systemic lupus erythematosus **infectious fever** **agranulocytosis** liver damage first stage of labor	0.25	0.157
dizziness cough grey sputum chest pain tightness in the chest shortness of breath vomiting	space-occupying lesion of the right lung	space-occupying lesion of the lung coronary heart disease pneumonia stage III hypertension disease lung infection	0.00	0.043

References

[1] R. Alizadehsani, J. Habibi, M.J. Hosseini, H. Mashayekhi, R. Boghrati, A. Ghandeharioun, B. Bahadorian, Z.A. Sani, A data mining approach for diagnosis of coronary artery disease, Comput. Methods Programs Biomed. 111 (1) (2013) 52–61.

[2] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512.

[3] M. Bastian, S. Heymann, M. Jacomy, et al., Gephi: an open source software for exploring and manipulating networks., ICWSM 8 (2009) 361–362.

[4] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, J.Mach.Learn.Res. 3 (Feb) (2003) 1137–1155.

[5] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. 2008 (10) (2008) P10008.

[6] C. Buckley, E.M. Voorhees, Evaluating evaluation measure stability, in: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2000, pp. 33–40.

[7] D.M. Chickering, D. Heckerman, C. Meek, Large-sample learning of bayesian networks is np-hard, J. Mach. Learn. Res. 5 (Oct) (2004) 1287–1330.

[8] J. Cong, H. Liu, Approaching human language with complex networks, Phys. Life Rev. 11 (4) (2014) 598–618.

[9] A.C. Constantinou, N. Fenton, W. Marsh, L. Radlinski, From complex questionnaire and interviewing data to intelligent bayesian network models for medical decision support, Artif. Intell. Med. 67 (2016) 75–93, doi:10.1016/j.artmed.2016.01.002.

[10] S.N. Dorogovtsev, A.V. Goltsev, J.F. Mendes, Critical phenomena in complex networks, Rev. Mod. Phys. 80 (4) (2008) 1275.

[11] P. Erds, A. Rnyi, On random graphs i, Publicationes Mathematicae 6 (1959) 290–297.

[12] D.D. Feng, Biomedical Information Technology, Academic Press, 2011.

[13] M.J. Flores, A.E. Nicholson, A. Brunskill, K.B. Korb, S. Mascaro, Incorporating expert knowledge when learning bayesian network structure: a medical case study, Artif. Intell. Med. 53 (3) (2011) 181–204.

[14] P. Fuster-Parra, P. Tauler, M. Bennasar-Veny, A. Ligza, A. Lpez-Gonzlez, A. Aguil, Bayesian network modeling: a case study of an epidemiologic system analysis of cardiovascular risk, Comput. Methods Programs Biomed. 126 (2016) 128–142, doi:10.1016/j.cmpb.2015.12.010.

[15] K.-I. Goh, I.-G. Choi, Exploring the human diseasome: the human disease network, Brief. Funct. Genomics (2012) els032.

[16] K.-I. Goh, M.E. Cusick, D. Valle, B. Childs, M. Vidal, A.-L. Barabasi, The human disease network, Proc. Natl. Acad. Sci. 104 (21) (2007) 8685–8690.

[17] T.J. Hannan, Electronic medical records, Health Inf. 133 (1996) 133–148.

[18] M. Hariharan, K. Polat, R. Sindhu, A new hybrid intelligent system for accurate detection of parkinson's disease, Comput. Methods Programs Biomed. 113 (3) (2014) 904–913.

[19] D.E. Heckerman, E.J. Horvitz, B.N. Nathwani, Toward normative expert systems: the pathfinder project, Methods Inf. Med. 31 (1991) 90I105.

[20] J.L. Herrera, R. Srinivasan, J.S. Brownstein, A.P. Galvani, L.A. Meyers, Disease surveillance on complex social networks, PLoS Comput. Biol. 12 (7) (2016) e1004928.

[21] i2b2, 2010 i2b2/va challenge evaluation assertion annotation guidelines, Available on https://www.i2b2.org/NLP/Relations/assets/Assertion%20Annotation%20Guideline.pdf (28 October) (2010b).

[22] i2b2, 2010 i2b2/va challenge evaluation concept annotation guidelines, Available on https://www.i2b2.org/NLP/Relations/assets/Concept%20Annotation%20Guideline.pdf (28 October) (2010a).

[23] U. Iqbal, C.-K. Hsu, P.A.A. Nguyen, D.L. Clinciu, R. Lu, S. Syed-Abdul, H.-C. Yang, Y.-C. Wang, C.-Y. Huang, C.-W. Huang, Y.-C. Chang, M.-H. Hsu, W.-S. Jian, Y.-C.J. Li, Cancer-disease associations: a visualization and animation through medical big data, Comput. Methods Programs Biomed. 127 (2016) 44–51, doi:10.1016/j.cmpb.2016.01.009.

[24] S. Kasakawa, T. Yamanishi, T. Takahashi, K. Ueno, M. Kikuchi, H. Nishimura, Approaches of Phase Lag Index to EEG Signals in Alzheimer's Disease from Complex Network Analysis, Springer International Publishing, Cham, pp. 459–468. doi:10.1007/978-3-319-23024-5_42.
[25] S. Keretna, C.P. Lim, D. Creighton, K.B. Shaban, Enhancing medical named entity recognition with an extended segment representation technique, Comput. Methods Programs Biomed. 119 (2) (2015) 88–100.
[26] A.N. Kho, J.A. Pacheco, P.L. Peissig, L. Rasmussen, K.M. Newton, N. Weston, P.K. Crane, J. Pathak, C.G. Chute, S.J. Bielinski, et al., Electronic medical records for genetic research: results of the emerge consortium, Sci. Transl. Med. 3 (79) (2011) 79re1–79re1.
[27] J.G. Klann, P. Szolovits, S.M. Downs, G. Schadow, Decision support from local data: creating adaptive order menus from past clinician behavior, J. Biomed. Inf. 48 (2014) 84–93.
[28] H. López-Fernández, M. Reboiro-Jato, D. Glez-Peña, F. Aparicio, D. Gachet, M. Buenaga, F. Fdez-Riverola, Bioannote: a software platform for annotating biomedical documents with application in medical learning environments, Comput. Methods Programs Biomed. 111 (1) (2013) 139–147.
[29] M. Madkour, D. Benhaddou, C. Tao, Temporal data representation, normalization, extraction, and reasoning: a review from clinical domain, Comput. Methods Programs Biomed. 128 (2016) 52–68.
[30] K. McGarry, K. Emery, V. Varnakulasingam, S. McDonald, M. Ashton, Complex Network Based Computational Techniques for 'Edgetic' Modelling of Mutations Implicated with Cardiovascular Disease, in: P. Angelov, A. Gegov, C. Jayne, Q. Shen (Eds.), Advances in Computational Intelligence Systems: Contributions Presented at the 16th UK Workshop on Computational Intelligence, September 7–9, 2016, Lancaster, UK, Springer International Publishing, Cham, 2017, pp. 89–106.
[31] S. Milgram, The small world problem, Psychol. Today 2 (1) (1967) 60–67.
[32] M.E. Newman, The structure and function of complex networks, SIAM Rev. 45 (2) (2003) 167–256.
[33] M.E. Newman, Power laws, pareto distributions and zipf's law, Contemp. Phys. 46 (5) (2005) 323–351.
[34] M. Pettorruso, A. Fasano, L.D. Risio, L. Ricciardi, M.D. Nicola, G. Martinotti, L. Janiri, A.R. Bentivoglio, Punding in non-demented parkinson's disease patients: relationship with psychiatric and addiction spectrum comorbidity, J. Neurol. Sci. 362 (2016) 344–347, doi:10.1016/j.jns.2016.02.016.
[35] H.-H. Rau, C.-Y. Hsu, Y.-A. Lin, S. Atique, A. Fuad, L.-M. Wei, M.-H. Hsu, Development of a web-based liver cancer prediction model for type ii diabetes patients by using an artificial neural network, Comput. Methods Programs Biomed. (2015).
[36] D. Scott, S.T. Dumais, G.W. Furnas, T.K. Lauer, H. Richard, Indexing by latent semantic analysis, in: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, 1999, pp. 391–407.
[37] R.A.A. Seoud, M.S. Mabrouk, Tmt-hcc: a tool for text mining the biomedical literature for hepatocellular carcinoma (hcc) biomarkers identification, Comput. Methods Programs Biomed. 112 (3) (2013) 640–648.
[38] E.H. Shortliffe, J.J. Cimino, Biomedical Informatics: Computer Applications in Health Care and Biomedicine, Springer Science & Business Media, 2013.
[39] Y. Tachimori, H. Iwanaga, T. Tahara, The networks from medical knowledge and clinical practice have small-world, scale-free, and hierarchical features, Physica A 392 (23) (2013) 6084–6089.
[40] J. Turian, L. Ratinov, Y. Bengio, Word representations: a simple and general method for semi-supervised learning, in: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2010, pp. 384–394.
[41] R. Van Der Hofstad, Random graphs and complex networks, Available on http://www.win.tue.nl/rhofstad/NotesRGCN.pdf (2009) 11.
[42] M. Velikova, J.T. van Scheltinga, P.J. Lucas, M. Spaanderman, Exploiting causal functional relationships in bayesian network modelling for personalised healthcare, Int. J. Approximate Reasoning 55 (1) (2014) 59–73.
[43] A. Vinayagam, T.E. Gibson, H.-J. Lee, B. Yilmazel, C. Roesel, Y. Hu, Y. Kwon, A. Sharma, Y.-Y. Liu, N. Perrimon, A.-L. Barabsi, Controllability analysis of the directed human protein interaction network identifies disease genes and drug targets, Proc. Natl. Acad. Sci. 113 (18) (2016) 4976–4981, doi:10.1073/pnas.1603992113.
[44] R.C. Wasserman, Electronic medical records (emrs), epidemiology, and epistemology: reflections on emrs and future pediatric clinical research, Acad. Pediatr. 11 (4) (2011) 280–287.
[45] D.J. Watts, S.H. Strogatz, Collective dynamics of 'small-world' networks, Nature 393 (6684) (1998) 440–442.
[46] J. Yang, Y. Guan, B. He, C. Qu, Q. Yu, Y. Liu, Y. Zhao, Annotation guidelines for named entities and entity relations of chinese electronic medical records, J. Softw. 3 (2016).
[47] J. Yang, Q. Yu, Y. Guan, Z. Jiang, An overview of research on electronic medical record oriented named entity recognition and entity relation extraction, Acta Autom. Sin. 40 (8) (2014) 1537–1562.
[48] W. Zhang, T. Yoshida, X. Tang, A comparative study of tf* idf, lsi and multi–words for text classification, Expert Syst. Appl. 38 (3) (2011) 2758–2765.
[49] X. Zhong, Q. He, J. Liao, X. Yin, G. Zhao, M. Li, The compatibility law of chinese patent medicines for the treatment of coronary heart disease angina pectoris based on association rules and complex network, Int. J. Clin. Exp. Med. 9 (6) (2016) 9418–9424.
[50] X. Zhou, L. Huang, W. Xiong, Key technology based on network pharmacology of complex networks, in: 2016 IEEE International Conference on Big Data Analysis (ICBDA), 2016, pp. 1–5, doi:10.1109/ICBDA.2016.7509826.
[51] X. Zhou, J. Menche, A.-L. Barabási, A. Sharma, Human symptoms–disease network, Nat. Commun. 5 (2014).

Publication 5

Clinical-decision support based on medical literature: A complex network approach

Jingchi Jiang, Jichuan Zheng, Chao Zhao, Jia Su, Yi Guan, and Qiubin Yu

Physica A 459 (2016) 42–54

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Clinical-decision support based on medical literature: A complex network approach

Jingchi Jiang [a], Jichuan Zheng [b], Chao Zhao [a], Jia Su [a], Yi Guan [a,*], Qiubin Yu [c]

[a] School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China
[b] RICOH Software Research Center (Beijing) CO., LTD, Beijing 100044, China
[c] Medical Record Room, The 2nd Affiliated Hospital of Harbin Medical University, Harbin 150086, China

HIGHLIGHTS

- We constructed a medical literature network (MLN) based on retrieved literature.
- The MLN improves the relevance retrieval result for clinical-decision support.
- We also proposed a re-ranking model to sort all retrieved literature by relevance.
- Our clinical-decision method based on the MLN yields higher scores in TREC 2015.
- Our study results confirmed that the MLN can facilitate the investigation of CDS.

ARTICLE INFO

Article history:
Received 28 December 2015
Received in revised form 7 April 2016
Available online 28 April 2016

Keywords:
Small-world
Scale-free
Complex network
Medical literature network
Clinical decision support

ABSTRACT

In making clinical decisions, clinicians often review medical literature to ensure the reliability of diagnosis, test, and treatment because the medical literature can answer clinical questions and assist clinicians making clinical decisions. Therefore, finding the appropriate literature is a critical problem for clinical-decision support (CDS). First, the present study employs search engines to retrieve relevant literature about patient records. However, the result of the traditional method is usually unsatisfactory. To improve the relevance of the retrieval result, a medical literature network (MLN) based on these retrieved papers is constructed. Then, we show that this MLN has small-world and scale-free properties of a complex network. According to the structural characteristics of the MLN, we adopt two methods to further identify the potential relevant literature in addition to the retrieved literature. By integrating these potential papers into the MLN, a more comprehensive MLN is built to answer the question of actual patient records. Furthermore, we propose a re-ranking model to sort all papers by relevance. We experimentally find that the re-ranking model can improve the normalized discounted cumulative gain of the results. As participants of the Text Retrieval Conference 2015, our clinical-decision method based on the MLN also yields higher scores than the medians in most topics and achieves the best scores for topics: #11 and #12. These research results indicate that our study can be used to effectively assist clinicians in making clinical decisions, and the MLN can facilitate the investigation of CDS.

* Correspondence to: School of Computer Science and Technology, Harbin Institute of Technology, Comprehensive Building 803, China. Tel.: +86 186 8674 8550.
E-mail address: guanyi@hit.edu.cn (Y. Guan).

http://dx.doi.org/10.1016/j.physa.2016.04.026
0378-4371/

1. Introduction

With the advancing age of populations worldwide, people have paid more attention to health problems. Each year, the numbers of deaths caused by cardiovascular diseases (CVDs) and hypertension are estimated to be 17.5 million and 7.1 million, respectively [1]. A World Health Organization (WHO) report on CVDs shows that 80% of all CVD deaths are due to heart attacks and strokes, which represent approximately 31% of all deaths globally [2]. Moreover, yearly figures from the WHO revealed that a person dies of diabetes and diabetic complications about every 10s. Diabetes is directly responsible for 1.5 million deaths in 2012 and 89 million disability adjusted life years [3]. The population of diabetic adults is expected to reach 300 million by 2025 [4]. Finding an effective auxiliary means to assist clinicians making a more correct clinical decision has become a critical problem to upgrade the clinician knowledge and reduce mortality.

Clinical-decision support (CDS) [5–9] which is a health information technology, provides physicians, patients, and other health professionals with knowledge and person-specific information, that is intelligently filtered and retrieved at appropriate times, to enhance patient health and health care [10]. In making clinical decisions, clinicians often seek out medical literature on how to best care for their patients. Medical literature can answer the three most common generic clinical questions faced by clinicians on a daily basis: "What is the patient's diagnosis?", "What tests should the patient receive?" and "How should the patient be treated?". However, given the volume of existing literature and the rapid pace at which new research is published, locating the most relevant and timely information for a particular clinical need can be a daunting and time-consuming task [11].

The Text Retrieval Conference (TREC) 2015 CDS track [12], which is similar to the goal of TREC 2014 [13–15], is designed to retrieve relevant medical literature to answer generic clinical questions on 30 actual patient records. The patient record typically describes three types of challenging medical cases and consists of 10 records per type, including diagnosis, test and treatment. Each record mainly contains two sections: description (detailing the patient condition) and summary (extracting meaningful information from the description based on the experience of doctors). The corpus for the retrieval task is the Open Access Subset of PubMed Central (PMC) on January 21, 2014, which contains a total of 733,138 literature [11]. According to the summary of a patient record, its description, or both, participants are challenged to retrieve a ranked set of 1000 papers at most, which are likely to support the decision of a physician on appropriate patient care.

The important aspects of the CDS according to medical literature have been discussed, and many valuable research ideas have been proposed. Garcia-Gathrighta et al. [16] adopted the vector space model using term frequency-inverse document frequency similarity and a unigram language model with Jelinek–Mercer smoothing. Mourao et al. [17] proposed multiple information retrieval techniques: retrieval functions, re-ranking, query expansion and classification of medical articles. Xu et al. [18] demonstrated the efficiency of the Johns Hopkins University HAIRCUT retrieval engine using character *n*-grams as an indexing term. Choi et al. [19] proposed an external tagged knowledge-based query expansion method for relevance ranking. Moreover, a machine-learning classifier-based text categorization method was used for the task-specific ranking.

Although many experts and scholars have explored the issues of clinical decision in different perspectives and realized a series of achievements, the use of complex network approach to solve the clinical decision problem has not yet been studied. In the present study, we focus on helping clinicians make better clinical decisions by retrieving relevant medical literature. In summary, we conducted our investigation in the following manner:

(1) We proposed a method of building a medical literature network (MLN). Then, we further analyzed the topological structure and characteristics of the MLN, which refers to the features of a complex network.
(2) According to the MLN and some analytical methods of complex networks, we adopted two strategies to mine potential literature, which can also assist clinicians making clinical decisions in addition to the basic literature retrieval.
(3) Combining the relevance factor of a search engine with the structural factor of the MLN, we further proposed a re-ranking model to sort all retrieved literature.
(4) From the comparison with those of other participants in TREC 2015, we numerically found that our approach can better improve the normalized discounted cumulative gain (NDCG) indicator than the median scores in most topics.

The rest of this paper is organized as follows: in Section 2, we introduce the structural characteristics of a complex network and the MLN. In Section 3, the potential-literature-mining algorithm and re-ranking model are proposed on the basis of a complex network approach. In Section 4, we further evaluate the validity of the potential-literature-mining and the accuracy of the re-ranking model. Finally, we conclude this paper and discuss directions for future work in Section 5.

2. Construction of the MLN

2.1. Process of relevant literature retrieval

To search some indicative literature to help clinicians make clinical decisions, we first need to retrieve some relevant literature using a search engine. Some classical retrieval techniques are adopted, including index building, query construction and literature retrieval.

The corpus from the PMC is given as a set of XML files. Therefore, an XML parser is employed to extract the PMC IDs, keywords, titles, abstracts, full texts and references. If an abstract is not available, the conclusion section will be used as a

substitute for the abstract. On the basis of the above-described work, the index files are created using the search engine called Indri.

The query construction consists of query extraction, query expansion and query set generation. In the process of query auto-construction, MetaMap (a tool to map biomedical text to the Unified Medical Language System (UMLS) Metathesaurus) is used to extract the medical concepts from the summary section of a patient record. We regard these filtered medical concepts as the basic query set. However, the basic queries, which are only extracted from the given patient record, cannot exactly describe the topic of each record. To compensate for the insufficiency of the basic queries, we further adopt the UMLS Metathesaurus to expand these concepts. After a series of steps, the query sets are generated in a format that conforms to Indri.

The organizers of TREC 2015 allow at most 1000 retrieved literature to be submitted for each patient record. We select the top 1000 papers as the result, which are ranked as the default score by the search engine.

2.2. Characteristics of the complex network

A complex network is composed of a large number of nodes and their intricate relationships. Complex network theories [20–23] have been employed to analyze many aspects of natural language processing, including language translation of a semantic network [24], investigation of knowledge graph [25], and construction of a literature citation network [26]. These complex networks frequently possess small-world [27] and scale-free [28] features. Small-world networks have a large clustering coefficient and a small average path length, which means that strong connectivity and high correlation exist between the nodes in the complex network. Scale-free networks are characterized by a power-law decay of the degree distribution. When some nodes fail, the topology of a complex network is more likely to be influenced than a random network [29].

According to the characteristics of a complex network, Tachimori et al. [30] constructed a hospital network and a medical knowledge network. An experimental study proved that the structure of a clinical practice may emerge from the mutual influence of medical knowledge and clinical practice. Therefore, we assume that an MLN can help clinicians in clinical decisions.

2.3. MLN

According to the analysis of the results of TREC 2014, we find that words often simultaneously occur in the annotated relevant literature. We believe that these simultaneously occurring words can reveal the relevance of the literature. To validate this assumption, we build an intuitive co-occurrence network based on 1000 retrieved literature. From the title, abstract, keyword and reference to literature, we empirically extract the co-occurring words using the top level of the MeSH hierarchy [17] as follows:

- Diagnosis: B03, B04, C
- Test: E01
- Treatment: D02, D04, D06, D26, D27, E02, E04.

When a common medical word from the above list appears on two papers, an edge will be created to connect them. Moreover, the edge weight gradually grows, along with increase in the number of common words. After the 1000 retrieved literature are iterated, we build an MLN based on the co-occurring words. This MLN is composed of the literature as the node and the co-occurring words as the edge. The topology of this network is shown in Fig. 1.

Furthermore, we use Gephi [31], which is an open-source graph visualization and manipulation software, to visualize the MLN. Using the modularity-class [32] algorithm to detect communities, the MLN is divided into several communities marking different colors. The community structure plays a vital role in mining the relevant literature. The features of this network include the following: (1) the literature nodes within the same community are strongly attached to one another [33,34], and (2) the nodes from different communities represent a "weaker" relationship [35].

2.4. Network analysis

To analyze the MLN structure, we present the degree and the graph distance distributions. First, we randomly select three different types of patient records from 30 medical cases. Because the number of nodes is fixed at 1000, the difference between these MLNs lies in the number of edges. In the diagnosis, test and treatment MLNs, the numbers of edges are 25,601, 42,413 and 61,113, respectively. Second, we show the degree distribution of the three MLNs in log–log plot. To some extent, the degree distribution of the three MLNs generally follows the truncated power-law distribution [36] in Fig. 2. Thus, the scale-free property of the MLN is proved.

Meanwhile, we show the graph distance distribution in Fig. 3, including the betweenness centrality, closeness centrality and eccentricity distributions.

Furthermore, the average path length and the average clustering coefficient for the three MLNs are calculated. Table 1 lists the diagnosis, test and treatment MLNs. C_{mln} and C_{random} are defined as the average clustering coefficients of the MLN and the corresponding random network, respectively. Similarly, L_{mln} is defined as the average path length. Because $L_{mln}/L_{random} \sim 1$

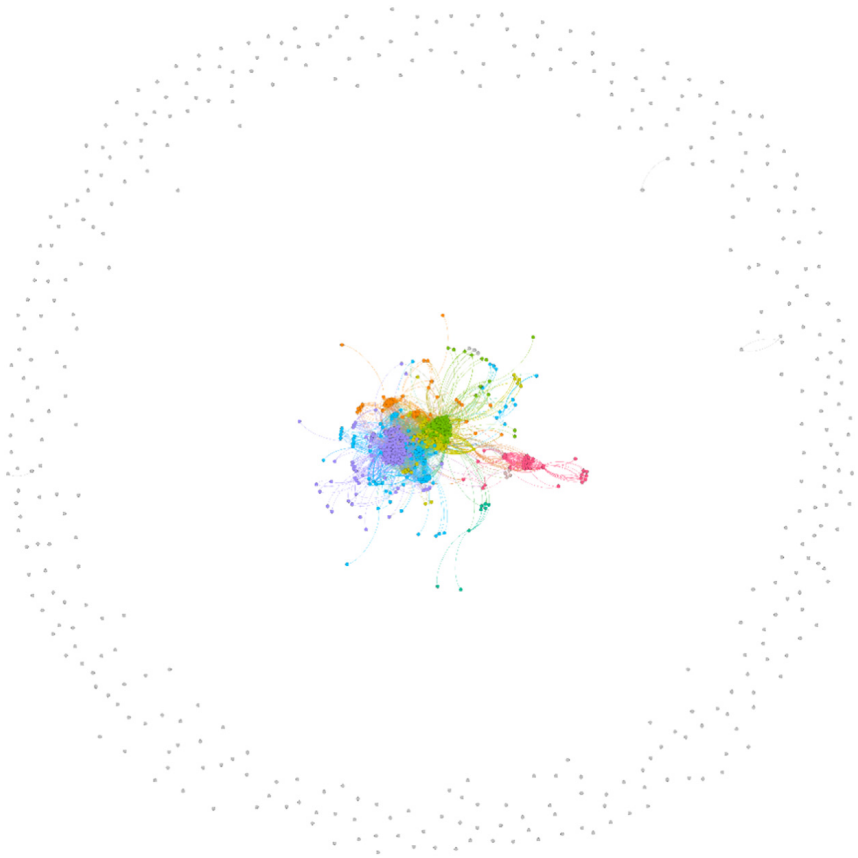


Fig. 1. The topology of MLN. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

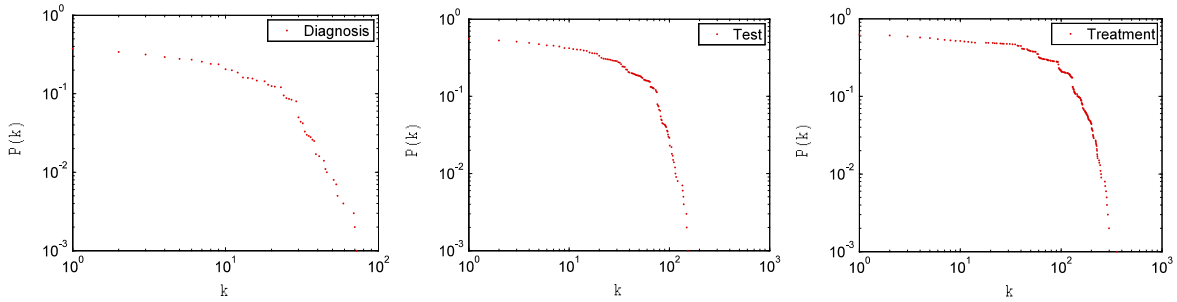


Fig. 2. Degree distribution of the diagnosis, test and treatment MLNs.

Table 1
Comparison of the MLN statistics with the corresponding random network.

Network	Nodes	Edges	C_{mln}	C_{random}	L_{mln}	L_{random}
$MLN_{diagnosis}$	1000	25,601	0.879	4.99×10^{-3}	2.29	2.02
MLN_{test}	1000	42,413	0.887	8.14×10^{-3}	2.17	1.92
$MLN_{treatment}$	1000	61,113	0.885	9.36×10^{-3}	1.97	1.89

and $C_{mln} \gg C_{random}$, the MLNs have almost the same average path length and a far larger average clustering coefficient compared with the corresponding random network. According to the small-world characteristics, the MLNs have small-world features.

In conclusion, these data demonstrate that the MLN possesses complex network properties: scale free and small world. Therefore, we can regard the MLN as a complex network.

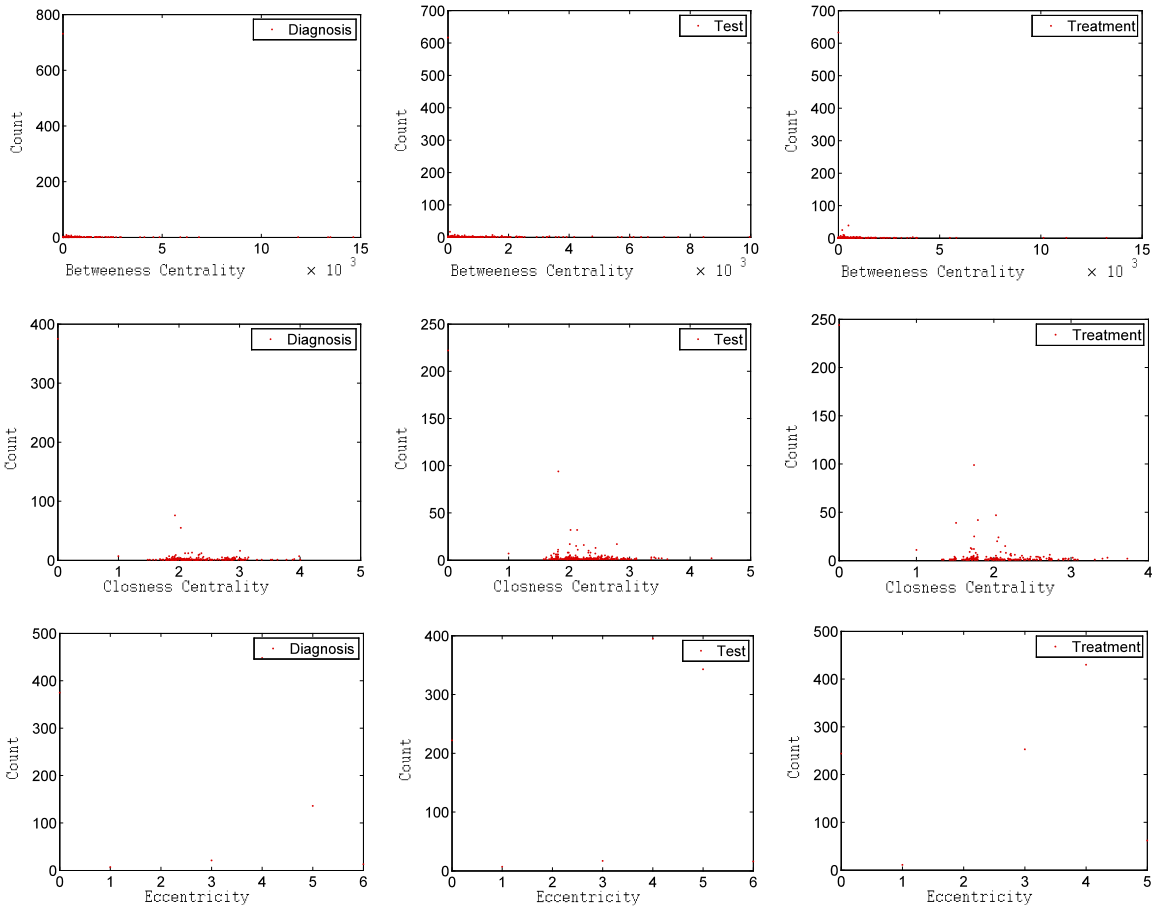


Fig. 3. Graph distance distribution of the diagnosis, tests and treatment MLNs.

3. Methodology

3.1. Mining of potential literature

Automatic extraction and expansion suffer from limitations and uncertainties, which lead to incomplete and non-credibility of the query set. Therefore, some relevant papers, which can also answer the clinical question, might be missed except for the 1000 retrieved literature. To solve this problem, we propose two different methods based on the MLN, to mine potential relevant literature from the rest of the corpus. The first method employs the characteristic of clustering coefficient [37–39] to identify whether a node belongs to potential relevant literature. The other method calculates the connectivity [40] between the specific community and node.

3.1.1. Mining of potential literature based on clustering coefficient

In some fields of a complex network, the related research of mining of potential nodes has been profoundly studied. In social networks, the node-mining technique is always applied in the community-detection algorithm and important-node perception algorithms.

In the decision file of TREC 2014, the relevant papers are labeled by medical librarians and physicians. Then, we find that most of the relevant papers are located inside the community in the MLN, whereas the discrete nodes always play the non or minimally relevant roles. Furthermore, we determine that every community has dissimilar emphases for a given patient record. The co-occurring medical words from a community are more suitable as a topic of the patient record, and the literature within this community is more relevant.

According to the above-mentioned analysis, we propose a node-mining method to identify the potential relevant literature on the basis of a clustering coefficient. Let $G = (V, E)$ denote an MLN, where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of n relevant papers, and $E \subseteq V \times V$ is a set of undirected co-occurring word relationships. The clustering coefficient can be divided into two categories: local and network-average clustering coefficients. The local clustering coefficient is defined as the proportion of connections among its neighbors, which are actually realized compared with the number of all possible connections. The local clustering coefficient of v_i in an undirected network is defined as follows:

$$L(v_i) = \{v_j : e_{ij} \in E\} \tag{1}$$

$$C(v_i) = \frac{2\left|\left\{e_{jk} : v_j, v_k \in L(v_i), e_{jk} \in E\right\}\right|}{k_i(k_i - 1)}. \tag{2}$$

The neighborhood $L(v_i)$ of node v_i represents its immediately connected neighbors. Parameter k_i is defined as the number of connections between node v_i and community ζ. The network average clustering coefficient is defined as the mean of the entire node coefficient within the community. $C(\zeta)$ represents the average clustering coefficient of community ζ:

$$C(\zeta) = \frac{\sum_{i=1}^{n} C(v_i)}{n}. \tag{3}$$

If the local clustering coefficient of potential node v_i is greater than the average clustering coefficient of specific community ζ, we can conclude that node v_i has high similarity to some nodes within ζ. Then, we locate node v_i into ζ as a potential relevant literature. Along with the continuous increase in the potential literature, some new MeSH terms will be obtained from the MLN that can precisely describe the topic. After all of the papers are traversed, a more comprehensive MLN $G' = (V', E')$ is built.

3.1.2. Mining of potential literature based on connectivity

In MLN, not all communities are suitable for the topic of a patient record. Choosing the relevant communities and identifying the potential nodes using the structure of these communities are very important. Considering the importance of the community structure in identifying potential nodes, we propose a node-mining method based on the community connectivity. First, we quantify the relevance of the community by calculating the MeSH lexical density, and choose a community with the highest lexical density.

$$\zeta = \underset{\varphi}{\operatorname{argmax}} \left\{ \frac{\left|\left\{term_{\varphi} : \varphi \subseteq G, term \in MeSH\right\}\right|}{\log\left|\{v_i : v_i \in \varphi\}\right|} \right\} \tag{4}$$

where $term_{\varphi}$ denotes a MeSH medical term in the community φ and ζ denotes the densest community. We assume that a community with high lexical density characteristics covers more healthcare areas and is likely to identify more relevant nodes. To avoid selecting some incorrectly small communities as ζ, we consider a logarithmic function to represent the community scale, and set $|v_i : v_i \in \varphi| > 1$. After the densest community is determined, we identify the potential nodes by centering on the structure of this community.

$$\frac{|\{v_m : v_m \in \zeta\}|}{\left|\left\{e_{ij} : v_i, v_j \in \zeta\right\}\right|} \leq \left|\left\{e_{jk} : v_j \in \zeta, v_k \notin \zeta, e_{jk} \in E\right\}\right|. \tag{5}$$

When the number of connections between v_k and ζ is greater than the community connectivity, we can indicate that node v_k is similar to the topic of community ζ. In conclusion, the two node-mining methods have all presented the importance of the network structure of the MLN. The effectiveness of these methods is verified through a series of experiments.

3.2. Re-ranking model

In this section, we will discuss how to re-rank the relevant literature on the basis of the MLN. After identifying the potential literature, the list of relevant literature contains more than 1000 papers. Therefore, we should uniformly re-rank the retrieved and potential literature and select the top 1000 papers to answer the clinical question.

In the process of determining the relevance of the literature, the average clustering coefficient of a community and the importance of a node are considered as a structural factor in the re-ranking model. The calculation of the average clustering coefficient of a community is similar to that of Eq. (3). In a complex network, the measures of the node importance are varied, such as PageRank, betweenness centrality and degree. In this study, we use these measures as the node importance and analyze the effectiveness of each measure in the experimental part.

In addition to the structural factor, relevance factor is also considered in the re-ranking model. The retrieved literature by the search engine has a default sort order with a relevance score. However, the absence of the relevance score for the potential literature makes it impossible to calculate the re-rank score in a uniform formula. Therefore, a computational method for calculating the relevance score of the potential literature is proposed, which is defined as follows:

$$Score(v_i) = \begin{cases} Score_{se}(v_i) & v_i \in \textit{ Retrieved Set} \\ \dfrac{C_{\zeta}(v_i) \cdot \sum_j Score_{se}(v_j)}{n} & v_i \in \textit{ Potential Set},\ v_j \in \zeta,\ e_{ij} \in E \end{cases} \tag{6}$$

$Score_{se}(v_i)$ represents the default relevance score of retrieved literature v_i, determined by a search engine. $C_{\zeta}(v_i)$ is the local clustering coefficient between node v_i and community ζ. $\sum_j Score_{se}(v_j)$ denotes that the sum of the relevance scores of all the retrieved nodes connected to v_i. Because $Score_{se}(v) \in [0, 1]$ and $C_{\zeta}(v) \in [0, 1]$, we can deduce that $Score(v) \in [0, 1]$.

Following the previous ideas of the re-ranking model, we combine the structural factor with the relevance factor to calculate the relevance. Let $ReRank\,(v_i, G) \in [0, 1]$ be the relevance of node v_i, which can be calculated as follows:

$$ReRank(v_i, G') = \alpha \cdot C(loc(v_i)) \cdot I(v_i) + \beta \cdot Score(v_i) \tag{7}$$

where α and β are two tunable parameters. $loc(v_i)$ denotes the community where node v_i is located. The functions C and I represent the average clustering coefficients of the community and the importance of the node, respectively. To direct the value range of α and β, we suppose that G' exists such that $\alpha + \beta > 1$. We consider that only three nodes v_1, v_2 and v_3 in G', and every node contains all queries. Thus, we obtain $I\,(v_1) = I\,(v_2) = I\,(v_3) = 1$, $v_1, v_2, v_3 \in \zeta = G'$, $e_{12}, e_{13}, e_{23} \in E'$ and $Score\,(v) = 1$. In addition, we can prove that $C\left(G'\right) = 1$ according to the fully connected network. Using Eq. (7), we obtain $ReRank\left(v_i, G'\right) = \alpha \cdot C\,(loc\,(v_i)) \cdot I\,(v_i) + \beta \cdot Score\,(v_i) = \alpha + \beta > 1$. However, the result goes against $ReRank\,(v_1, G) \in [0, 1]$. Therefore, we have $\alpha + \beta \leq 1$. To avoid being trapped in the local optimum of the parameters, we choose a special case $\alpha + \beta = 1$, which can be transformed to $\beta = 1 - \alpha$. Using the method of potential-literature mining and re-ranking the literature, we design an algorithm to calculate the relevance of each literature in the MLN. The detailed algorithm is listed in Table 2.

According to the description of the mining of the potential-literature method, we must walk through the full corpus. To reduce the time complexity, the clinical-decision algorithm extracts the MeSH medical terms from the basic MLN and retrieves the c-top literature on the basis of each term by Indri. Then, these retrieved literature will displace the full corpus as the candidate of potential literature. The reasons are that the potential literature must be connected to the existing literature within the MLN and the edges are formed by the MeSH medical terms. Thus, the optimization algorithm only needs to traverse the occurring terms in the MLN. The algorithm results in time complexity $O(c \cdot k)$, where k is the number of MeSH terms.

4. Experiments and discussion

As participants to the TREC 2015 CDS track, we download 30 topics from the TREC official website [12], which consist of ten "diagnosis" topics, ten "test" topics and ten "treatment" topics. Meanwhile, the TREC official provides the same corpus with TREC 2014, which contains a total of 733,138 literature. To train our model, we regulate the values of some parameters and compare the effectiveness of two mining methods using the 30 topics of TREC 2014. After the optimal model is constructed, we use this model to complete the TREC 2015 CDS track. From the comparison with those of the other participants, our model achieves better result in TREC 2015. The three main modules of the model are shown as follows:

4.1. Mining method analysis

In this section, we focus on the effectiveness of mining of the potential literature. We adopt two methods, namely, based on clustering coefficient (CC-based) and based on connectivity (CO-based) mining methods, to analyze the corpus of TREC 2014. Fig. 4 shows the x-axis representing the serial numbers of topics from 1 to 30. The first 10 topics belong to the "diagnosis" type. The middle ten topics belong to the "test" type. The remaining ten topics belong to the "treatment" type. The y-axis represents the amount of relevant literature. By referring to the decision file of TREC 2014, we can determine which papers are relevant. We choose the number of mined potential literature as 50, 100, and 200 respectively. The experimental results of the CC-based and CO-based mining methods are shown in Fig. 5.

Fig. 5 shows the basic results representing the number of relevant literature, which were retrieved by a search engine. We can observe that the basic results are unstable. The highest result reaches 211 and the lowest is zero. Following the increase in the number of basic results, the number of potential relevant literature increases under the CC-based method. When the effectiveness of the basic results is maintained at a low level, the performance of the CC-based method is unsatisfactory. A similar situation is observed for the CO-based method. These results illustrate that the effect of the mining method depends on the established MLN. The reason for this is that when a irrelevant MLN is built, the potential literature which are identified based on the structure of the MLN, will inevitably result in an off topic. In addition, the sequence of the mining effect for different types of topic from high to low is from treatment to test and to diagnosis. Finally, we compare the effect of the two methods. Fig. 5 shows that the CC-based method is superior to the CO-based method.

To further verify whether the proportion of relevant paper position in all literature will decrease with the increase in the number of mined literature, we calculate the ratio of the different scales of mined literature. We choose the number of mined literature (NUM) as 50, 100, and 200 respectively. The results are shown in Fig. 6.

Under the CC-based method, the proportion of the relevant paper remains steady when NUM $=$ 50, 100 and 200. In particular, when NUM $=$ 200, the ratio is higher than the basic results in some topics, which illustrates that node addition by the CC-based method is effective. In contrast, we can observe that a greater NUM will result in a smaller relevant ratio under the CO-based method.

In conclusion, according to the MLN, the effectiveness of the CC-based method is better than that of the CO-based method. The number of relevant papers obviously increase under the CC-based method. In addition, the proportion of relevant papers remains steady even when NUM $=$ 200. Therefore, the CC-based method with NUM $=$ 200 is very significant for mining more relevant papers.

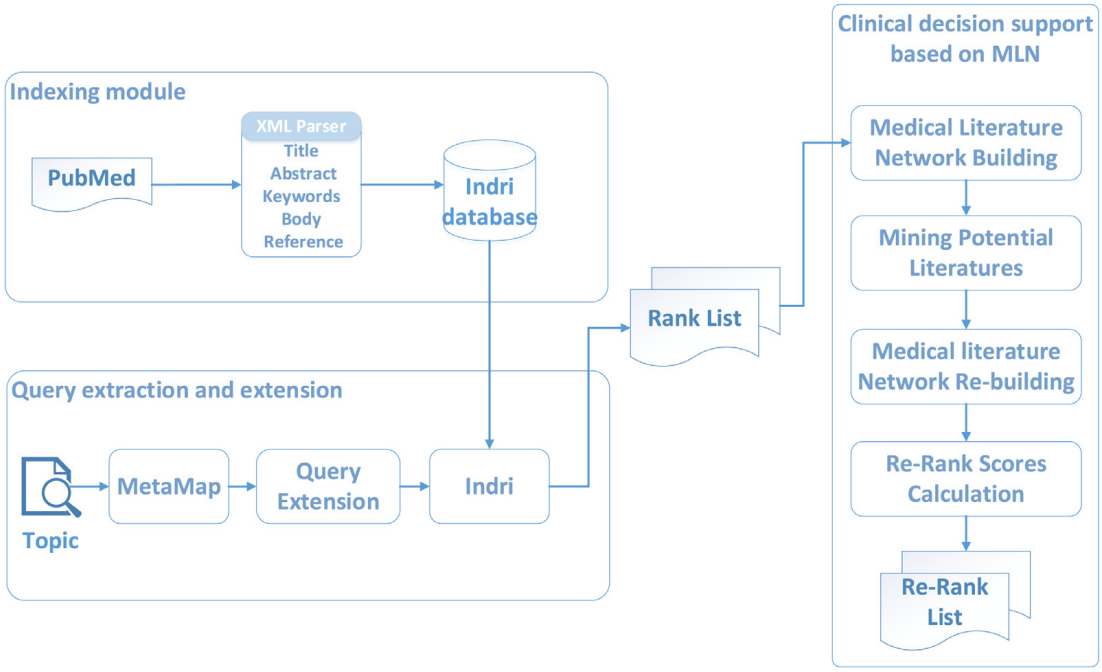


Fig. 4. Flow diagram of the CDS based on MLN.

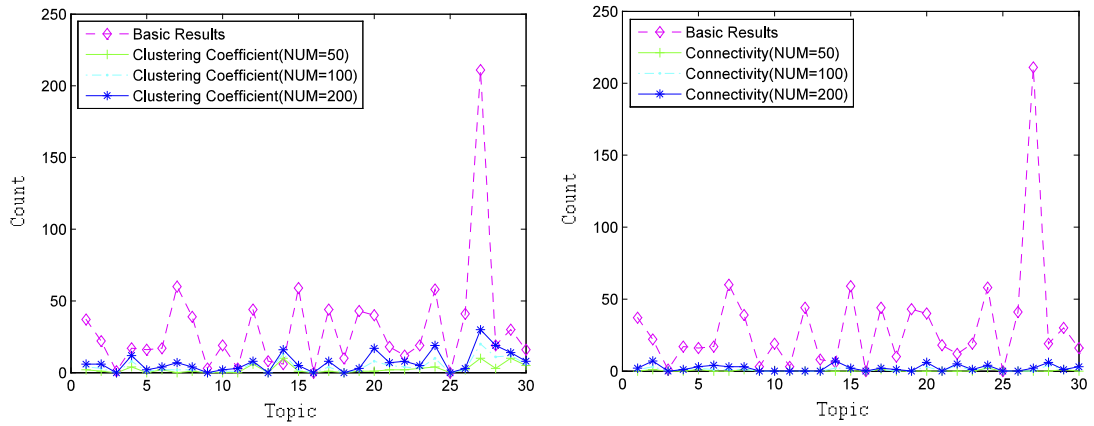


Fig. 5. Number of relevant literature under the CC-based and CO-based methods.

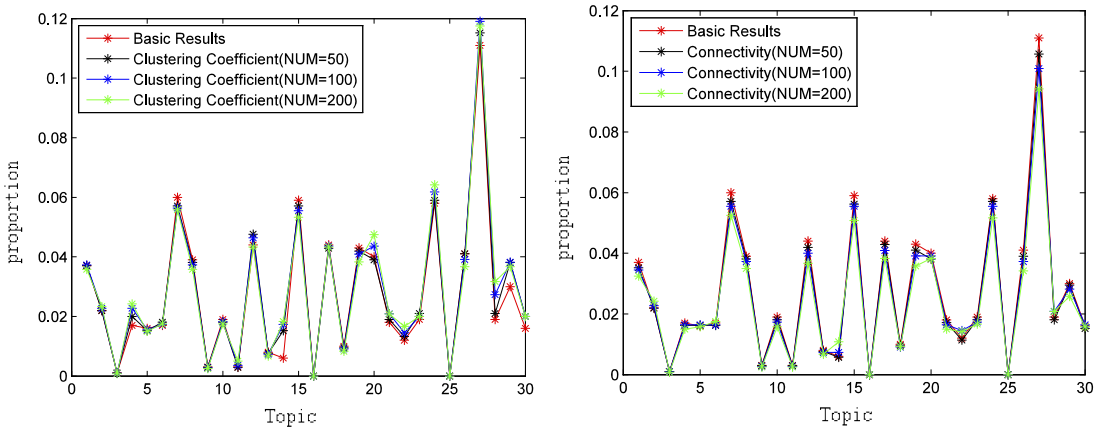


Fig. 6. Proportion of relevant literature under the CC-based and CO-based methods.

Table 2
CDS algorithm based on medical literature.

Algorithm 1: CDS algorithm based on medical literature

Input: R_i: the content of the i-th patient record.

Output: $List_{fin}$: the list of relevant literature after re-ranking.

Begin

1: Initialize the MeSH medical terms $List_{mesh}$ and the number of retrieved literature by Indri, c.

2: Initialize parameters α, β.

3: Extract the queries from the summary section of R_i by MetaMap → $Query = \{Query_1, Query_2, ..., Query_n\}$

4: Extend the basic $Query$ by UMLS, $Query$ → $Query' = \{Query_1', Query_2', ..., Query_m'\}$

5: ***Function*** $Indri(Query', 1{,}000)$ ***do*** → $List_{ini}$

6: Initialize the MLN $G_{ini} = \{V_{ini}, E_{ini}\}$ based on $List_{ini}$ in accordance with $List_{mesh}$.

7: Extract the MeSH medical terms from G_{ini}, $List_{mesh}'$.

8: ***for*** $mesh_i \in List_{mesh}'$ ***do***

9: $List_{temp} = Indri(mesh_i, c)$

10: ***for*** $node_{temp} \in List_{temp}$ ***do***

11: ***if*** the clustering coefficient method is adopted ***then***

12: Calculate the clustering coefficient of communities, C_{com}.

13: Calculate the clustering coefficient of $node_{temp}$, C_{node}.

14: ***if*** $C_{node} > C_{com}$ ***then*** $node_{temp} \rightarrow G_{ini}$

15: ***else if*** the connectivity method is adopted ***then***

16: Calculate the lexical density of communities, and select the most densely community $Com_{max.}$

17: Calculate the connectivity of Com_{max} and $node_{temp}$ → C_{com}, C_{node}.

18: ***if*** $C_{node} > C_{com}$ ***then*** $node_{temp} \rightarrow G_{ini}$

19: ***end if***

20: ***end for***

21: ***end for***

22: After identifying the potential literature, the more comprehensive MLN is built, $G_{ini} \rightarrow G_{fin}$

23: ***Function*** $Re\text{-}rank(G_{fin}, \alpha, \beta) \rightarrow List_{fin}$ ***do***

24: return $List_{fin}$

4.2. Re-ranking effect analysis

After the potential-literature mining, we adopt the re-ranking model to identify 1000 relevant literature from the 1200 literature. However, the re-ranking model contains some uncertain factors that may influence the accuracy, such as weight parameter α and the importance of node $I(v_i)$. To choose appropriate α and the proper importance index of the node, we calculate the NDCG of each topic to evaluate the effects under different α values and importance indexes. The NDCG formula [41] for measuring the relevance of the retrieval result is expressed as follows:

$$NDCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i} \tag{8}$$

where rel_i represents the score of the ith literature; relevant is one, whereas irrelevance is zero. p is the number of the retrieved result, which is 1000 in this paper. We randomly select six topics from "diagnosis", "test" and "treatment". The

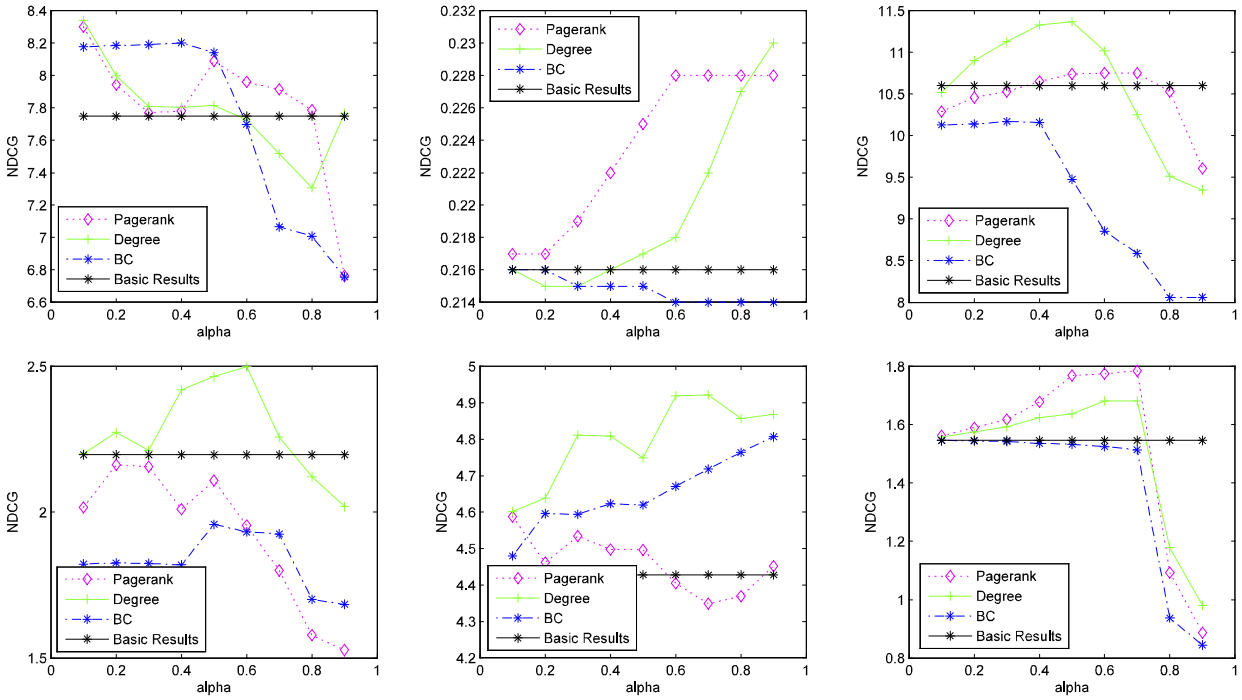


Fig. 7. Re-ranking effect analysis of the "diagnosis" topics.

x-axis represents weight parameter α from 0.1 to 0.9. The y-axis represents the NDCG score. In addition, we employ PageRank, degree and betweenness centrality as the importance indexes. The NDCG of the basic results without re-ranking is also calculated for comparison of the effects. The experimental results are shown in Figs. 7–9.

Fig. 7 shows that the curve of the NDCG score is irregular. When the value of α is between 0.4 and 0.6, the NDCG generally shows an increase, which illustrates that the re-ranking model exhibits the best performance when the structural and relevance factors coexist and play the same important roles. We can also observe that the re-ranking results of the degree distribution is better than the two other importance indexes. Finally, we numerically select 0.6 as weight parameter α and the degree distribution as the importance index of the re-ranking model for the "diagnosis" topics.

Following the same analysis of the mean of the "diagnosis" topic, we choose the weight parameter and importance index for the "test" topics and "treatment" topics. However, some singular curves appear in the abovementioned topics. The fourth topic in Fig. 9 shows that the curve does not change with α and remains at zero because our method does not find any relevant literature in this topic. Furthermore, we find that some topics exhibit continual decrease with increasing α, which illustrates that the accuracy of the re-ranking model decreases with the increasing percentage of the structural factor. Therefore, we doubt that a less comprehensive MLN has been built, and the structural factor of the re-ranking model suffers from a negative effect. Considering these singular curves and all the cases, we use 0.3 and 0.7 as the weight parameter of the "test" topics and "treatment" topics, respectively. PageRank and betweenness centrality are employed as the importance indexes, respectively.

In addition, the distribution of the relevant literature is analyzed. We perform statistical analysis of the number of relevant literature in the top 100, 200, 500 and 1000 retrieved results. Fig. 10 shows that the number of relevant literature is evenly distributed. This phenomenon does not show that all the papers are concentrated in a specific range, such as distributed between 500 and 1000. Therefore, this experiment proves the rationality of the re-ranking model.

Given the above results, we can conclude that the re-ranking model can improve the performance of relevance ranking. Through many experiments and analyses, we choose an appropriate α value and a proper importance index for the different types of topics. Finally, we reveal the distribution of relevant literature in the retrieval results and verify the rationality of the re-ranking model.

4.3. Comparing the submitted runs with those of the other participants

As participants in TREC 2015, our retrieval results, which are generated by the mining method and re-ranking model, are submitted to the official website. After the evaluation results of our runs are announced, a set of experiments for comparison with those of the other participants is given, as shown in Fig. 11.

Fig. 11 shows that the performance of our method based on the MLN is superior to the median scores in most topics. In addition, the sequence of the mining effect for the different types of topics from high to low is as follows: treatment, test and diagnosis. The result is similar to the experimental results using the topic of TREC 2014. Surprisingly, we find that

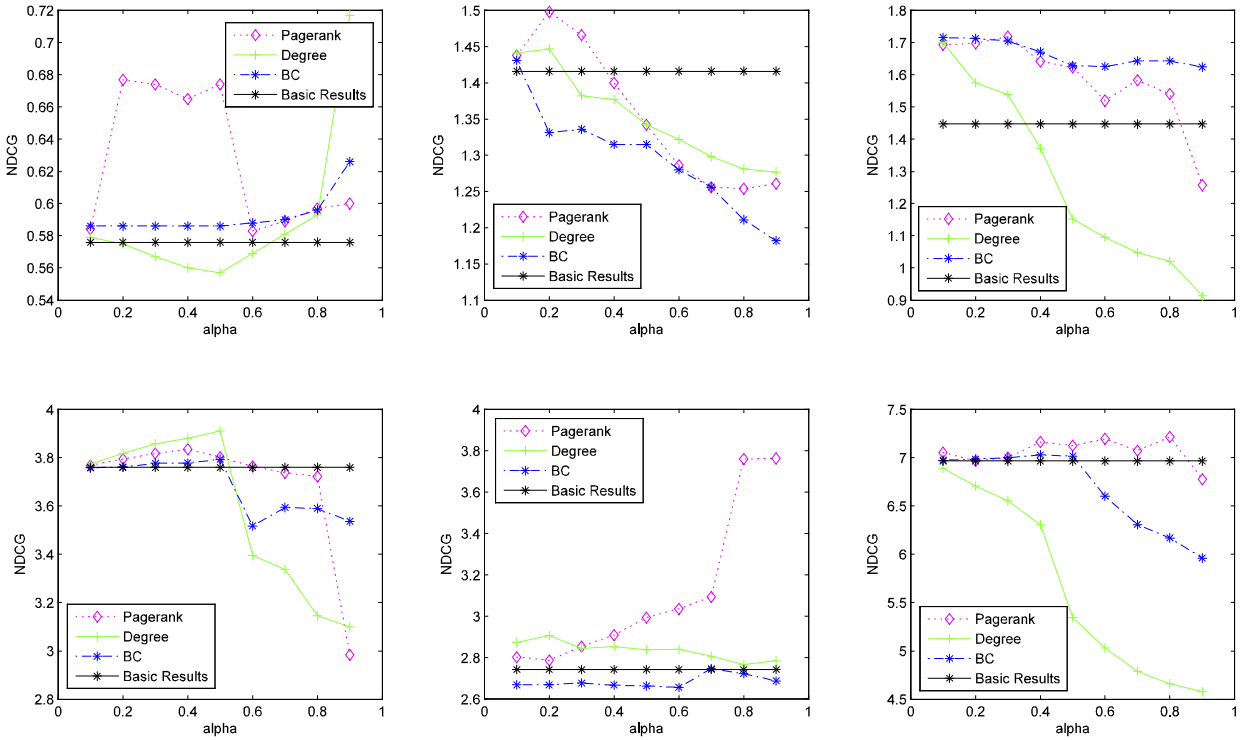


Fig. 8. Re-ranking effect analysis of the "test" topics.

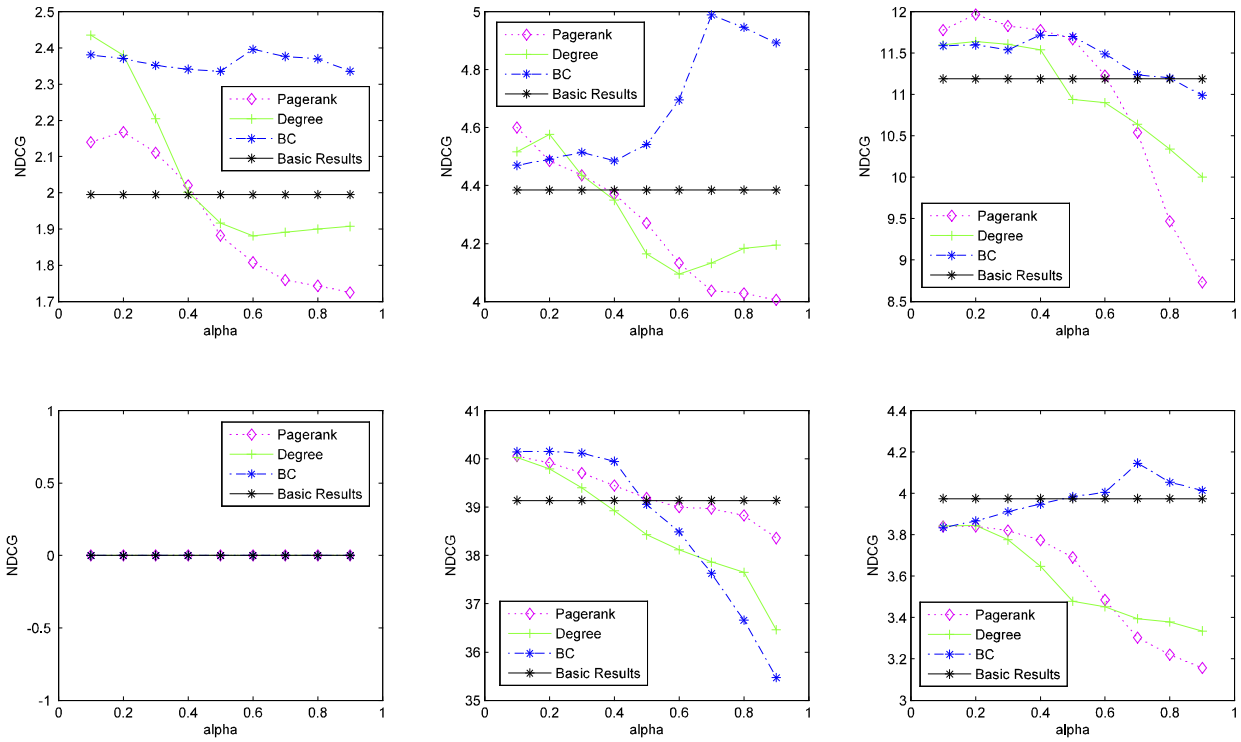


Fig. 9. Re-ranking effect analysis of the "treatment" topics.

our method achieves the best score in two topics: #11 and #12. These results further testify to the effectiveness of the CDS algorithm.

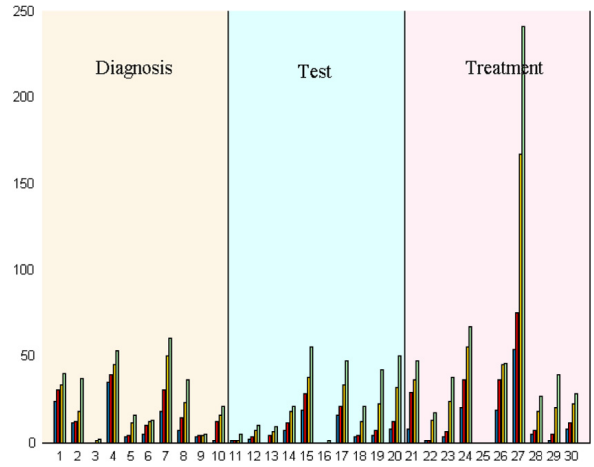


Fig. 10. Distribution of the relevant literature.

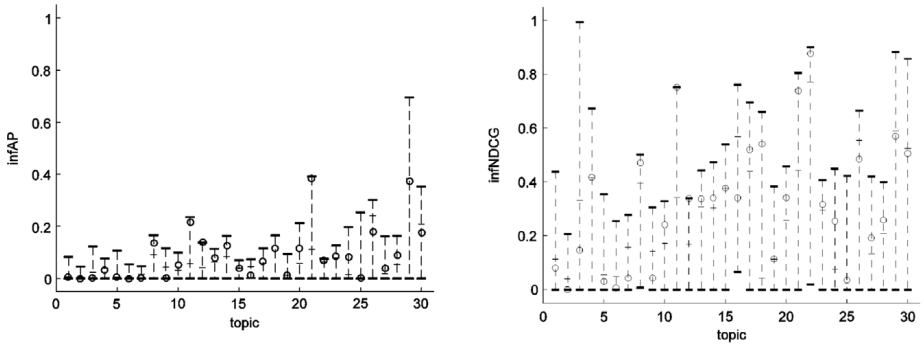


Fig. 11. Comparison of our submitted runs with those of the other participants.

5. Conclusion and future work

In this paper, we presented our study on CDS on the basis of medical literature. We built an MLN using the basic retrieved results. The small-world and scale-free properties were validated in the MLN. To help clinicians make the most appropriate diagnosis, test and treatment, we used the features of a complex network to mine more papers. We adopted two mining methods, namely, CC-based and CO-based methods, to identify the potential literature. We further proposed the re-ranking model, which synthesizes the structural factor of the MLN and the relevance factor of the search engine. We numerically determined that the re-ranking model can improve the relevance of the results. Compared with those of the other participants in TREC 2015, the effectiveness of our method was verified.

In the future, according to the different sources of medical knowledge, we will construct and study the massive commonsense knowledge network to more effectively help clinicians in making clinical decisions.

Acknowledgments

The Open Access Subset of PubMed Central used in this paper was provided by TREC 2015 Clinical Decision Support (CDS) track. We would like to thank the reviewers for their detailed reviews and insightful comments, which have helped to improve the quality of this paper.

References

[1] World Health Organization (WHO), Cardiovascular diseases. [Online] Available: http://www.who.int/cardiovascular_diseases/en/.
[2] American Heart Association, Make the Effort to Prevent Heart Disease. [Online] Available: http://www.heart.org/HEARTORG/GettingHealthy/Make-the-Effort-to-Prevent-Heart-Disease-with-Lifes-Simple-7_UCM_443750_Article.jsp.
[3] World Health Organization (WHO), Raised fasting blood glucose. [Online] Available: http://www.who.int/gho/ncd/risk_factors/blood_glucose_text/en/.
[4] C. Day, The rising tide of type 2 diabetes, Br. J. Diabetes Vasc. Dis. 1 (1) (2001) 37–43.
[5] M.A. Musen, B. Middleton, R.A. Greenes, Clinical decision-support systems, in: Biomedical Informatics, Springer, London, 2014, pp. 643–674.
[6] T.J. Bright, A. Wong, R. Dhurjati, et al., Effect of clinical decision-support systems: A systematic review, Ann. Intern. Med. 157 (1) (2012) 29–43.
[7] C.C. Tseng, P.J. Gmytrasiewicz, Real-time decision support and information gathering system for financial domain, Physica A 363 (2) (2006) 417–436.
[8] T.J. Carney, G.P. Morgan, J. Jones, et al., Using computational modeling to assess the impact of clinical decision support on cancer screening improvement strategies within the community health centers, J. Biomed. Inform. 51 (2014) 200–209.

[9] P.S. Roshanov, N. Fernandes, J.M. Wilczynski, et al., Features of effective computerised clinical decision support systems: Meta-regression of 162 randomised trials, BMJ 346 (2013) f657.
[10] J.A. Osheroff, J.M. Teich, B. Middleton, et al., A roadmap for national action on clinical decision support, J. Am. Med. Inform. Assoc. 14 (2) (2007) 141–145.
[11] M.S. Simpson, E. Voorhees, W. Hersh, Overview of the TREC 2014 clinical decision support track, in: Proc. 23rd Text Retrieval Conference (TREC 2014), National Institute of Standards and Technology(NIST), 2014.
[12] TREC Clinical Decision Support Track, 2015 Task: Generic Clinical Questions. [Online] Available: http://www.trec-cds.org/2015.html.
[13] TREC Clinical Decision Support Track, 2014 Task: Generic Clinical Questions. [Online] Available: http://www.trec-cds.org/2014.html.
[14] H.S. Oh, Y. Jung, KISTI at TREC 2014 Clinical Decision Support Track: Concept-based Document Re-ranking to Biomedical Information Retrieval, TREC 2014 Track, 2014.
[15] J. Gobeillab, A. Gaudinata, E. Paschec, et al. Full-texts representation with Medical Subject Headings, and co-citations network rerank-ing strategies for TREC 2014 Clinical Decision Support Track, TREC 2014 Track, 2014.
[16] J.I. Garcia-Gathrighta, F. Menga, W. Hsua, UCLA at TREC 2014 Clinical Decision Support Track: Exploring Language Models, Query Expansion, and Boosting, TREC 2014 Track, 2014.
[17] A. Mourao, F. Martins, J. Magalhaes, NovaSearch at TREC 2014 Clinical Decision Support Track, TREC 2014 Track, 2014.
[18] T. Xu, P. McNamee, D.W. Oard, HLTCOE at TREC 2014: Microblog and Clinical Decision Support, TREC 2014 Track, 2014.
[19] S. Choi, J. Choi, SNUMedinfo at TREC CDS track 2014: Medical case-based retrieval task, TREC 2014 Track, 2014.
[20] K. Yamamoto, Y. Yamazaki, Structure and modeling of the network of two-Chinese-character compound words in the Japanese language, Physica A 412 (2014) 84–91.
[21] D. Tomasi, N.D. Volkow, Resting functional connectivity of language networks: Characterization and reproducibility, Mol. Psychiatry 17 (8) (2012) 841–854.
[22] Y. Gao, W. Liang, Y. Shi, et al., Comparison of directed and weighted co-occurrence networks of six languages, Physica A 393 (2014) 579–589.
[23] C. Yi, Y. Bao, J. Jiang, et al., Modeling cascading failures with the crisis of trust in social networks, Physica A 436 (2015) 256–271.
[24] D.R. Amancio, M.G.V. Nunes, O.N. Oliveira, et al., Using metrics from complex networks to evaluate machine translation, Physica A 390 (1) (2011) 131–142.
[25] S. Guo, Q. Wang, B. Wang, et al. Semantically smooth knowledge graph embedding, in: Proceedings of ACL, 2015.
[26] D.R. Amancio, O.N. Oliveira, L. da Fontoura Costa, Three-feature model to reproduce the topology of citation networks and the effects from authors' visibility on their h-index, J. Informetr. 6 (3) (2012) 427–434.
[27] D.J. Watts, S.H. Strogatz, Collective dynamics of "small-world" networks, Nature 393 (6684) (1998) 440–442.
[28] H. Jeong, B. Tombor, R. Albert, et al., The large-scale organization of metabolic networks, Nature 407 (6804) (2000) 651–654.
[29] A.L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1998) 509–512.
[30] Y. Tachimori, H. Iwanaga, T. Tahara, The networks from medical knowledge and clinical practice have small-world, scale-free, and hierarchical features, Physica A 392 (23) (2013) 6084–6089.
[31] M. Bastian, S. Heymann, M. Jacomy, Gephi: an open source software for exploring and manipulating networks, in: Proc. of 3rd International AAAI Conference on Weblogs and Social Media, ICWSM, 2009.
[32] M.E.J. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. 103 (23) (2006) 8577–8582.
[33] L. Šubelj, M. Bajec, Unfolding communities in large complex networks: Combining defensive and offensive label propagation for core extraction, Phys. Rev. E 83 (3) (2011) 036103.
[34] J. Shang, L. Liu, X. Li, et al., Epidemic spreading on complex networks with overlapping and non-overlapping community structure, Physica A 419 (2015) 171–182.
[35] J. Xie, S. Kelley, B.K. Szymanski, Overlapping community detection in networks: The state-of-the-art and comparative study, ACM Comput. Surv. (CSUR) 45 (4) (2013) 43.
[36] M.E.J. Newman, The structure and function of complex networks, SIAM Rev. 45 (2) (2003) 167–256.
[37] T. Zhou, G. Yan, B.H. Wang, Maximal planar networks with large clustering coefficient and power-law degree distribution, Phys. Rev. E 71 (4) (2005) 046141.
[38] J. Saramäki, M. Kivelä, J.P. Onnela, et al., Generalizations of the clustering coefficient to weighted complex networks, Phys. Rev. E 75 (2) (2007) 027105.
[39] Y. Cui, X. Wang, J. Li, Detecting overlapping communities in networks using the maximal sub-graph and the clustering coefficient, Physica A 405 (2014) 85–91.
[40] L. Wu, Q. Tan, Y. Zhang, Network connectivity entropy and its application on network connectivity reliability, Physica A 392 (21) (2013) 5536–5541.
[41] E. Yilmaz, E. Kanoulas, J.A. Aslam, A simple and efficient sampling method for estimating AP and NDCG, in: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2008.

Publication 6

Classification of entities via their descriptive sentences

Chao Zhao, Min Zhao, and Yi Guan

Classification of entities via their descriptive sentences

Chao Zhao
Harbin Institute of Technology
zhaochaocs@gmail.com

Min Zhao
Baidu Inc.
zhaomin@baidu.com

Yi Guan
Harbin Institute of Technology
guanyi@hit.edu.cn

Abstract

Hypernym identification of open-domain entities is crucial for taxonomy construction as well as many higher-level applications. Current methods suffer from either low precision or low recall. To decrease the difficulty of this problem, we adopt a classification-based method. We pre-define a concept taxonomy and classify an entity to one of its leaf concept, based on the name and description information of the entity. A convolutional neural network classifier and a K-means clustering module are adopted for classification. We applied this system to 2.1 million Baidu Baike entities, and 1.1 million of them were successfully identified with a precision of 99.36%.

1 Introduction

Entity classification aims to determine the type (e.g., person, location) of a certain entity. It is important for several natural language processing (NLP) tasks, such as question answering, textual entailment, machine reading, and text summarization. It is also a crucial resource to introduce the concept-level features to enhance the generalization power of machine learning systems (Paulheim and Fümkranz, 2012).

There are two main methods for entity classification: named entity recognition and classification (NERC) and entity hypernym extraction. The former assumes that the type of an entity can be determined from its context. The classification tags utilized in NER systems are often coarse and do not involve all types of entities. Fine-grained NER faced with the difficulty of creating sufficient training data. The latter method can obtain extremely fine-grained categories of one entity, but it has to rely on an ontology to further obtain more general concepts. It also requires the co-occurrence of the entity and its hypernym in one sentence, which can make the recall lower. If there is no hypernym word in the context of the entity, and the context does not contain much information to determine the classification, these two methods could not work well, and we have to rely on external knowledge sources.

In this paper, we propose another method for entity classification. Rather than inferring the type of an entity from the context, we classify it according to its descriptive sentences. A descriptive sentence is used to describe certain attributes of an entity, and is a natural source for humans to recognize the type of the unknown entity. For example, from the sentence "The logo of Baidu is a bear paw.", we have no idea what *Baidu* is through the context information. But from a description of *Baidu*:

> Baidu, incorporated on 18 January 2000, is a Chinese web services company

We can easily conclude that *Baidu* is a COMPANY.

Motivated by this idea, we propose a classification-based method for entity classification. We first pre-define a hierarchical concept taxonomy, and then try to classify an entity to a leaf of this taxonomy, based on its description and name information. Our contributions are three-fold:

- We propose a simple classification-based method for entity classification, based on the description and name of the entity.
- We introduce a clustering module to alleviate the data noise and imbalance problem during training, as well as to select the highly confident predicted entities from the prediction set to improve the precision.
- We utilize this architecture on 2.1 million open-domain entities to verify its efficiency. Approximately 1.1 million entities are successfully identified, with a precision of 99.36%.

2 Methods

This section presents the dataset and the method used for entity classification.

2.1 Dataset and task

The entities and their corresponding descriptions utilized are from Baidu Baike. It contains nearly 15 million Chinese pages until now, which is much larger than that of Wikipedia (nearly 1 million). 12.4 million entities out of 15 have been linked to the Baidu Concept Base, a concept taxonomy, by the features from keywords, tags, or key-value tables, with a precision of 98%. We call them as *known entities*. Another 2.6 million *unknown entities* had not been linked because of the lack of or low quality of these features. The original model is therefore helpless for these entities, yet we want to identify their hypernyms according to their descriptions.

Linking entities to existing concept base can be naturally regarded as a classification task. We first manually select a mini concept taxonomy from the Baidu concept base, and regard the leaf nodes as classification labels. This operation decreases the number of classes to make the classification easier. As shown in left part of Table 1, the mini-taxonomy contains 48 classes. These concepts are selected because they are necessary and fine-grained enough to support subsequent applications. They cover more than 98.6% of the known entities in Baidu Baike.

2.2 Pipeline

The workflow of our method is shown in Figure 1, which contains three main modules: pre-processing, model training, and post-processing.

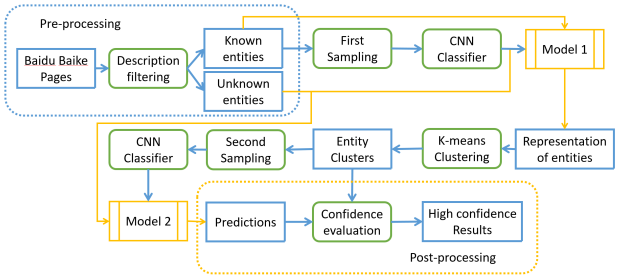


Figure 1: The workflow of the identification system.

Pre-processing is introduced to select entities that contain descriptions. Kazama and Torisawa (2007) directly regard the first sentence of an entity page as its description. However, we estimate from a sampled data of Baidu Baike and find that only approximately 84% of entity pages hold such assumption. To increase this percentage, we only regard the first sentence of an entity page as its description when it meets at least one of two requirements:

- The first sentence begins with the entity name (the title of entity page); or
- The head of the first sentence in its dependency tree is a verb phrase;

The unsatisfied entity pages are directly filtered. After filtering, the precision of the descriptions is increased to 97%. Nearly 9.7 million and 2.1 million entities are reserved from known and unknown entity sets, respectively, which constitute our training and prediction set. The training set has two problems that need to be addressed. First, since the precision of current entity-hypernym relationships is approximately 98%, the other 2% should be regarded as the noise, which would lead to performance decline for a non-specially designed model (Zhu and Wu, 2004). Second, since different concepts (classes) correspond to different numbers of entities (see Table 1), the training set is imbalanced, which would cause poor performance in minority classes. Instances in minority classes would also suffer from higher noise.

We adopt a standard convolutional neural network (CNN) classifier (Kim, 2014) for text classification. To solve the noise and imbalance problem, we first train the CNN model using small-sized balanced data. Based on the trained model, we obtain the vector representation of all training and predict entities. Then, we use a clustering-based re-sampling method to select the highly confident entities from the total training data, to obtain another large-sized training set. In this set, the noise and imbalance problems are alleviated. We re-train the CNN classifier accordingly and obtain the final classification model, which is used to predict the hypernyms of unknown entities.

To increase the precision of the prediction, we introduce a post-processing module to select the highly confident predicted entities. The details of the CNN classifier, the clustering and re-sampling module, and the post-processing module are described below.

2.3 CNN classifier

As shown in Figure 2, the CNN classifier has two input channels: the character-level name information and the word-level descriptive sentences, which relies on a word segmenter. We add special <start> and <end> symbols to both sides of the entity name, to help the CNN model capture the prefix and suffix features. The name information is removed from the description, to avoid feature redundancy. If the name is surrounded by the title mark (≪≫), a special Chinese punctuation to indicate the titles of books, films, and more, it would be moved along with the title to the name channel.

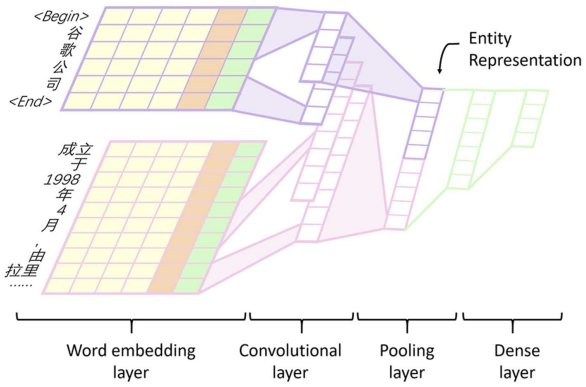


Figure 2: The architecture of the CNN classifier. The input of the name channel is “Google Inc.”, while that of the description channel is “established in April, 1998 by Larry...”.

The embedding of words has dimensionality of 200, which are pre-trained from Baidu Baike articles using the skip-gram model (Mikolov et al., 2013). The character-level and word-level embedding tables are shared, because the single character word is common in Chinese. Since the skip-gram model has a threshold to ignore less frequent words, these words do not have pre-trained embedding and would be assigned to random embedding. A common way to enhance the word embedding is to concatenate their corresponding part-of-speech (POS) and syntax embedding (Wang and Chang, 2016; Chen and Manning, 2014). These

features are reported to be beneficial to the hypernym identification. We therefore concatenate them to the word embeddings (shown in Figure 2 with different colors). The POS tagger, dependency parser, and the earlier segmenter are supported by the Baidu NLP Cloud service. Followed by a convolutional layer, a max-pooling layer, and two dense layers, we obtain the probability distribution of the input entity over the 48 classes.

2.4 Clustering and re-sampling

We first sample a relatively small number of entities randomly from each class and train a CNN model based on this small balanced set. The output of the pooling layer is a concatenated vector, and can be regarded as the representation of the entity. We expect that the entities in the same class would be close in the representation space. In other words, when we run a clustering algorithm over these entities, the cluster partition should be similar with the inherent classifications. If the percentage of a class in one cluster is tiny (e.g., less than $p\%$), we would suspect the entities in this class as noise and remove them. This would cause false filtering, but it would be better to remove several correctly labeled entities, rather than keep the falsely labeled ones. In this way, part of noise can be discarded.

The imbalanced problem is still alleviated with the under-sampling strategy. Balanced training data is fair for minority classes, while imbalanced data is better for the majority. To compromise, the size of sampled data in each class is still correlated with that of entire training data, but the proportion is smoothed by the logarithmic function. We first set a reference sample size N. For one class c, we denote the number of entities in c by P_c before noise filtering, and Q_c after filtering, where $P_c \geq Q_c$. There are three conditions (All the entities below refer to those belonging to class c):

1. If $P_c > N$ and $Q_c \geq N$, we would completely sample $\hat{N}_c = N \times (1 + \log_{10}(\frac{Q_c}{N}))$ entities from c;
2. If $P_c > N$ but $Q_c < N$, we rank the clusters in descending order by the number of filtered entities in c, and recall these filtered entities in order to make Q_c exactly larger or equal to N. It then becomes the first condition;
3. If $P_c < N$, we directly take all the entities (filtered and remained) to the sampled data.

The final problem is the sampling method. It relies on the clustering results we obtained. Instances in the large-sized class would be partitioned into many small clusters. If we under-sample in the class-level, we have the risk of not sampling entities from small-sized clusters, and may lead to the wrong classification for the predict entities in the same cluster. We therefore sampled from the cluster level in a proportional way, which can better guarantee that the sampled subset can cover all clusters. An example is shown in Figure 3, which compares the results of sampling from the class or cluster level.

2.5 Post-processing

To add predicted data into current taxonomy, we have to guarantee the precision benchmark of 95%, which is hard to achieve. We must select the highly confident predicted entities from the entire prediction set. Similar to the methods to detect noise data, this process also relies on the clustering results.

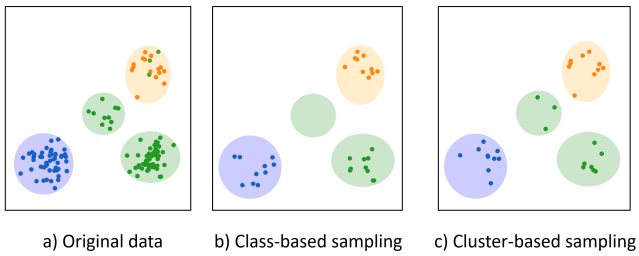


Figure 3: Resampling from the clusters. The ellipses represent the clusters and the color refers to the class. The noise is first removed from the original data. The class-level sampling fails to sample data from the center cluster, while the cluster-level sampling avoids this situation. The sampling process is simulated via the random.sample() method in Python.

In most cases, one cluster has both known and unknown entities. As shown in Figure 4, we can easily obtain the class distribution of known entities. The motivation of the confidence evaluation is intuitive: If most of the known entities are labeled as ★, and a test entity in the same cluster is also labeled as ★ by the classifier, then we have high confidence that this entity is correctly labeled. We grade the predict confidence of an unknown entity as four levels, according to the percentage of this class in the known entities: L1 (>99%), L2 (99%-50%), L3 (50%-5%), and L4 (<5%). If there are only test entities in a cluster, we simply label the confidence of a test entity as L5 if there are more than 50% entities labeled as the same class with it, or L6 otherwise.

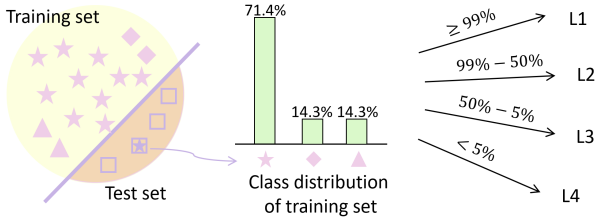


Figure 4: The confidence evaluation according to the class distribution of known entities. The unknown entities are represent with □, while other symbols represent the classes of known entities. Symbol in the □ is the prediction result of the unknown entity.

3 Experiment

3.1 The first classifier

To train the first classifier with the small balanced dataset, we randomly sampled 1,000 entities from each class, and split the training/validation set with ratio of 0.7/0.3 (the

numbers of entities are 33,600/14,400). Since parameter tuning was the only purpose, there was no separate test set. The structure and parameter were set as follows, according to the performance of the validation set:

The convolutional layer used filters with the window size as 1, 2, 3, 4, and each size contained 60 feature maps for every channel. The dimension of the pooling layer, therefore, was 480, and that of the hidden layer was 200. Two dropout layers with rates of 0.5, as well as an L_2-norm constraint for dense-layer parameters were utilized for regularization. Rectified linear units (ReLUs) were adopted as the activation function of the convolutional layer and the first dense layer. The softmax cross entropy was used as the loss function. The optimizer adopted the Adam algorithm Kingma and Ba (2014) with a learning rate of 0.001 and a mini-batch size of 500. The performance of the trained classifier on each class of the validation set is shown in Table 1. All macro P, R, and F_1 measures are 0.88.

We evaluated the influence of POS and syntax features on the final performance. The introduction of POS features increased the macro-F_1 from 87.95% to 88.36%. However, the continued concatenation of dependency tagging features decreased the measure to 88.06%. Similar results were reported by Sang and Hofmann (2009). It may be caused by the relatively low precision of the Chinese dependency parser. Therefore, we only concatenated the POS tagging features to the word embedding.

We also tried to change CNN to long short-term memory networks (LSTMs), to combine the word and character sequences as representation vectors. We found that, neither replacing CNN with LSTM on the name channel nor description channel can cause a large slump of macro-F_1 measure by more than 10%. A possible reason is that the LSTMs without the attention mechanism focus more on the last part of the sentence, where the information for hypernym identification may not exist.

3.2 Clustering

With the help of the classifier, we obtained the vector representation of all entities, with the dimensionality of 480. We run the K-means algorithm to partition these entities, and K is arbitrarily set as 10,000. The statistical quantities of these clusters are shown in Figure 5. Figure 5 a) shows that the number of clusters which possess more than 1,000 entities is roughly equal to that of clusters with less than 1,000 entities. Figure 5 b) and c) indicate that approximately one half of the clusters are almost pure, in which more than 99% of the entities belong to the same class.

We further analyzed the clusters of different classes. For each class, we select the clusters where this class has the most number of entities, and draw the boxplot of these numbers. The results are shown in Figure 6. The large-sized classes are more likely to be partitioned into large clusters. This indicates that despite the large numbers, the descriptions of these entities in the same class are not arbitrary. They instead have high similarity, and therefore the classification-based method is reasonable.

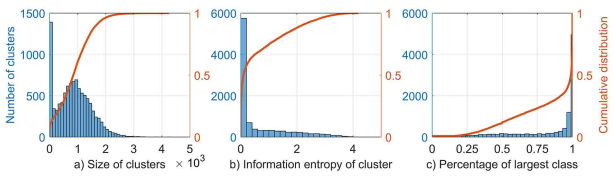


Figure 5: a) and b) show the distribution of size and information entropy of clusters, and c) shows the distribution of the percentage of the largest class in one cluster. All of the left y-axes represent the number of clusters, and right y-axes represent the corresponding cumulative distribution.

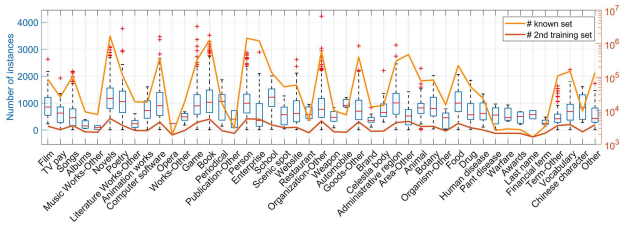


Figure 6: The boxplot described above. We also draw the number of entities in the known set, as well as the second training set for comparison. These refer to the right log-scaled axis.

3.3 The second classifier

We re-sampled training entities from these clusters using the strategy described in Section 2.4. Compared with the previous training set, the size of this re-sampled set enlarged from 33,600 to 154,479 (see Table 1). It can better represent the entire training set. Based on this set, we re-trained the second classifier from scratch, and then applied it to predict the hypernym of unknown entities. The post-processing module then partitioned these entities into six confidence levels. The precision of the predicted results was evaluated manually, by sampling a small subset. For simplicity, we refer to a confidence level of one class as a *group*. We randomly selected at least 40 entities from each group and labeled the predicted hypernyms as right or wrong. More than 10,000 entities are evaluated in total. In particular, entities being classified as "Other" would be linked to the higher-level concepts. For example, if a film is classified as VIDEOWORKS-OTHER, we would regard it as a VIDEO WORK and therefore labeled this prediction as correct.

The precision of each class c in the entire predict set was estimated by two ways:

1. Estimation at the class level: we sampled m entities from class c for manual evaluation in all, and r of them were right identified. The precision was calculated by $p_1 = r/m$;
2. Estimation at the group level: we denote by n the size of class c. Its i-th confidence level L_i has n_i entities with precision of p_i, which was estimated by the first way. Then $p_2 = (\sum_i p_i \cdot n_i)/n$

Since the evaluation subset was not uniformly sampled, we regarded P_2 as the primary estimation of the class-level precision. The size of some groups was less than 100, which would make small contributions to the expansion of our taxonomy. Therefore, we did not evaluate them.

We tried to correct the wrongly labeled entities while evaluating but found it difficult and time-consuming in practice. Therefore, the recall of the results could not be evaluated. Fortunately, we were more concerned about the precision, and the absence of recall was acceptable.

4 Results and discussion

4.1 Results

Table 1 shows the details of the precision of each group. Generally speaking, the precision declines from level 1 to level 4. Level 1 has a perfect performance, because they are selected from the almost pure clusters. The number of entities belonging to this level, as a result, is not very much, except several classes like ENTERPRISE, PERSON and POETRY. Instances of other concrete classes concentrated mainly on level 2 and level 3. There are still 49.7% of the classes in level 2 with a precision of more than 94%, and this percentage decreases to 8.7% in level 3. The special "Other" classes, which contain the remaining concepts not listed in the taxonomy, center on the fourth level with a relatively lower precision. It is because these "Other" classes contain more messy concepts, and are unlikely to be clustered into the relatively pure groups.

Instances in the fifth and sixth levels come from the clusters consisting of only predict entities, and there are two situations. If it is an accidental consequence caused by the K-means algorithm, the precision of this cluster would be affected by its nearest clusters. If it is caused because no similar entities existing in the known set, the precision would be low.

Finally, we choose the groups in which the precision is more than 94%, and add these entities to our existed taxonomy. 1.1 million out of 2.1 million are selected in this way, and the percentage is approximately 55%. According to the second precision estimation method, our selected subset has the precision of 99.36%.

4.2 discussion

The first sentence of an instance page from the web encyclopedia is an ideal source for the entity categorization task. The page editor would write the first sentence more clearly. In many cases, it is a standard definition sentence. On the other hand, when the editors try to write a new article, they may refer to the existing pages of the same kind of instances, leading to the similar form of sentences among these instances.

Since the precision of the knowledge base is more important, we designed the post-processing module to divide the predicted results into different confidence levels, and evaluate each level separately. In essence, it is a strategy to assign confidence to predicted instances according to the support number of similar training instances. The largest contribution to the high precision comes from the ENTERPRISE type, in which the names of entities have obvious suffix features (e.g., Inc.) and therefore a simple classifier can obtain satisfactory results. However, even we do not consider these entities, the precision of prediction set before and after the post-processing is 80.72% and 98.76%, respectively, which can still show the effectiveness of confidence-based filtering to the high precision.

The prerequisite of the classification strategy is the predefined taxonomy, which makes the categorization process more controllable compared with the hypernym extraction methods, but also restricts the granularity of the hypernym concept. How fine-grained should an identified category be depends on the subsequent tasks. Consider *Phenobarbital* as an example. For a medical ontology which is used for clinical support, it should be described as a hypnotics/anticonvulsants or in more detail. In other applications, like the intension detection of medical QA web, it is enough for the system to know it is a medicine, and therefore the coarse-grained categorization is acceptable.

4.3 Limitations

Despite the application background, it would be better to find a taxonomy that can compromise the granularity of the hypernym and the classification precision. In the clusters, we find that several clusters in the same class correspond to different sub-classes. The PERSON class, for example, is partitioned as ACTOR, SINGER, WRITER, and more, which shows the potentiality of the classifier for finer-grained classification.

Another limitation is the selection of descriptive sentences. Instead of regarding the first sentence of the Baidu Baike pages as the description, it is better to train a discriminator to select high-quality descriptions from web text. In this way, the entity set would not be restrained to those involved in Baidu Baike.

We utilized a simple CNN text classifier and flattened the hierarchical structure of the taxonomy during classification. More elaborate operations should further improve the performance.

5 Related works

5.1 Hypernym extraction

Hypernym extraction is utilized to find the hypernym of a certain entity from the sentences where the hypernym and the entity occur together. For example, from the sentence "... such authors as Herrick and Shakespeare." we can derive that "Herrick" and "Shakespeare" are authors. Such methods are mainly based on the lexical or syntactic patterns. They were designed manually at the beginning by Hearst and Hearst (1992), and then were mined automatically using machine learning techniques (Snow, Jurafsky, and Ng, 2004; Ritter, Soderland, and Etzioni, 2009; Kozareva and Hovy, 2010). These methods require the co-occurrence of hypernym-hyponym pairs in one sentence, and often suffer from relatively lower precision or recall (Seitner et al., 2016), because of the arbitrariness of the natural language.

Taxonomy			validation set			predict set			level 1		level 2		level 3		level 4		level 5		level 6	
			P	R	F	#num	P_1	P_2	#num	P	#num	P	#num	P	#num	P	#num	P	#num	P
Works	Video works	1. Film	0.81	0.95	0.88	8568	0.72	0.79	571	1	5764	0.94	883	0.32	1347	0.4			3	
		2. TV play	0.89	0.9	0.9	3425	0.89	0.87			2086	0.94	835	0.9	504	0.5				
		3. TV program	0.88	0.99	0.93	5404	0.61	0.73			363	0.52	4053	0.77	985	0.62			3	
		4. Other	0.59	0.54	0.56	6831	0.77	0.77							6823	0.77			8	
	Music works	5. songs	0.91	0.93	0.92	10271	0.86	0.97	3072	1	6378	1	603	0.65	218	0.33				
		6. albums	0.86	0.94	0.9	1575	0.73	0.80			940	1	471	0.52	163	0.45			1	
		7. Other	0.79	0.72	0.76	27436	0.62	0.56							27423	0.56	5		8	
	Literature works	8. novels	1	1	1	8806	0.82	0.93	2302	1	5807	0.97	594	0.48	98				5	
		9. poetry	0.91	0.97	0.94	8660	0.88	0.94	3531	1	3693	0.94	913	0.71	519	1			4	
		10. Other	0.76	0.68	0.72	30141	0.61	0.60					21455	0.65	8686	0.49				
	11. Animation works		0.88	0.85	0.87	6266	0.73	0.70	5		2741	1	1401	0.61	2116	0.38			3	
	12. Computer software		0.9	0.95	0.93	12670	0.74	0.63	203	1	6274	0.87	4892	0.35	1294	0.5			7	
	13. Opera		0.9	0.79	0.84	6362	0.68	0.55			1418	1	1115	0.52	3825	0.39			4	
	14. Other		0.86	0.68	0.76	15957	0.39	0.33							14760	0.3	898	0.77	299	0.4
15. Game			0.94	0.94	0.94	12562	0.78	0.72	1200	1	5214	1	4991	0.32	1156	0.9			1	
Public-ations	16. Book		0.78	0.89	0.83	34347	0.62	0.77	6718	1	21072	0.77	4956	0.45	881	0.92	119	1	601	0.6
	17. Periodical		0.94	0.98	0.96	2264	0.81	0.89	6		1609	0.94	230	0.65	418	0.83			1	
	18. Other		0.93	0.81	0.86	2827	0.71	0.71							2826	0.71			1	
19. Person			0.86	0.95	0.9	289215	0.76	0.94	215230	1	63179	0.78	8751	0.55	1964	0.65	11		80	
Organizations	20. Enterprise		0.98	0.93	0.96	678962	0.68	0.99	624168	1	6043	0.94	2268	0.65	9993	0.94	35830	0.88	660	0.84
	21. School		0.95	0.98	0.97	6397	0.8	0.90	2275	1	2116	1	1221	0.55	769	0.88			16	
	22. Scenic spot		0.83	0.95	0.89	15836	0.96	0.91	192	1	2951	1	10418	0.87	2271	1			4	
	23. Website		0.92	0.94	0.93	22456	0.62	0.66	16		12792	0.84	4743	0.45	4898	0.39			7	
	24. Restaurant		0.91	0.94	0.93	13521	0.97	0.96			3047	1	7409	0.97	3065	0.89				
	25. Other		0.83	0.69	0.75	79195	0.69	0.96							77129	0.97	1855	0.45	211	0.36
Goods	26. Weapon		0.9	0.96	0.93	8435	0.85	0.76	56		2334	0.9	2920	0.61	3113	0.82			12	
	27. Automobile		0.92	0.94	0.93	5949	0.79	0.69			1294	0.94	2772	0.71	1878	0.5			5	
	28. Other		0.82	0.86	0.84	94750	0.85	0.85							94700	0.85			50	
29. Brand			0.84	0.85	0.84	25332	0.72	0.70			6733	0.97	9821	0.68	8768	0.52			10	
30. celestial body			0.99	0.98	0.98	6799	0.98	0.99	3322	1	1048	0.94	775	1	1654	1				
Area	31. Administrative region		0.97	0.92	0.95	13447	0.69	0.96	8975	1	3131	1	632	0.71	538	0.5	138	1	33	
	32. Other		1	1	1	94760	0.94	0.92							94653	0.92	78		29	
Orga-nism	33. Animal		0.86	0.96	0.91	18658	0.95	0.89	341	1	10685	0.94	5146	0.97	2479	0.5			7	
	34. Botany		0.88	0.89	0.88	22962	0.82	0.75	204	1	10939	0.9	7987	0.65	3818	0.5			14	
	35. Other		0.91	0.8	0.85	6723	0.94	0.94							6721	0.94			2	
36. Food			0.91	0.89	0.9	12063	0.83	0.80	1515	1	6908	0.9	1975	0.65	1662	0.4			3	
37. Drug			0.92	0.87	0.89	23576	0.7	0.60	585	1	11459	0.84	8150	0.29	3377	0.5			5	
38. Chemical substance			0.8	0.9	0.84	21296	0.97	0.97					5983	0.97	15306	0.97			7	
39. Human disease			0.93	0.9	0.92	10844	0.61	0.51	20		4544	0.74	3984	0.39	2292	0.26			4	
40. Plant disease			0.97	0.99	0.98	659	0.81	0.79	1		519	1	40		99					
41. Warfare			0.97	0.96	0.96	2198	0.47	0.49			930	0.71	430	0.26	838	0.36				
42. Awards			0.98	0.98	0.98	2654	0.73	0.82			1728	0.97	714	0.55	208	0.48			4	
43. Last name			0.96	0.92	0.94	2281	0.83	0.78			585	0.87	678	0.81	1018	0.71				
Term	44. Financial term		0.88	0.96	0.92	8157	0.99	0.96			857	1	5379	0.97	1909	0.94			12	
	45. Other		0.74	0.64	0.69	328667	0.59	0.65					121277	0.91	187041	0.49	20092	0.64	257	0.38
46. Vocabulary			0.83	0.86	0.85	99821	0.9	0.92	7813	1	59575	0.99	23002	0.81	9396	0.71			35	
47. Chinese character			0.94	0.95	0.94	4386	1	0.99			4336	1	5		45					
48. Other			0.62	0.58	0.60															

Table 1: The statistical quantities and evaluation results of each class. From left to right, this table shows 1) the pre-defined concept taxonomy; 2) the P, R, and F measure of the first classifier on the validation set; 3) the number of entities predicted by the second classifier, as well the two estimated precision; and 4) the numbers and precision information of each group. The red shadow is a visualization of some important evaluation measures, while the green shadow indicates the percentage of the predicted entities in each confidence level.

5.2 Named entity recognition and classification

NERC was first introduced at the Sixth Message Understanding Conference (MUC-6) Grishman and Sundheim (1996). Early tasks focused most on three kinds of entity types: person, location, and organization. The classes were further fine-grained in subsequent works. For example, Fleischman and Hovy (2002) classified the person into sub-classes like politics and artist. In the field of biomedical informatics, researchers would focus more on the identification of specific terms, like proteins and DNAs (Leaman, Gonzalez, and Others, 2008). Nadeau (2007) presented a hierarchical structure for named entities, and the number of entity types is approximately 200. Ling and Weld (2012) defined 112 fine-grained tags and labeled the training data with the help of anchor links from Wikipedia text, then utilizing the perceptron as the classifier to determine the type of entities. The dominant supervised methods for NERC extract features from tokens themselves and their contexts, and no exterior knowledge source is involved.

6 Conclusion

This work offers a classification-based method for entity categorization. Based on the name and description information of an entity, we can classify it to one of the types via a CNN classifier. A clustering module is designed for noise filtering, training set sampling, and confidence evaluation for predicted results. We applied this method to 2.1 million open-domain entities, and 1.1 million are successfully classified with a precision of 99.36%, demonstrating the efficiency of our method.

In the future, we hope to develop a discriminator to find the entity descriptions in broader knowledge sources, and to identify the type of an entity from multiple descriptions.

References

Chen, D., and Manning, C. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 740–750.

Fleischman, M., and Hovy, E. 2002. Fine grained classification of named entities. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, 1–7. Association for Computational Linguistics.

Grishman, R., and Sundheim, B. 1996. Message Understanding Conference-6: A Brief History. In *COLING*, volume 96, 466–471.

Hearst, M. a., and Hearst, M. a. 1992. Automatic Acquisition of Hyponyms ftom Large Text Corpora. *Proceedings of the 14th conference on Computational Linguistics* 2:23–28.

Kazama, J., and Torisawa, K. 2007. Exploiting Wikipedia as External Knowledge for Named Entity Recognition. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (March):698–707.

Kim, Y. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kozareva, Z., and Hovy, E. 2010. A semi-supervised method to learn and construct taxonomies using the web. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, 1110–1118. Association for Computational Linguistics.

Leaman, R.; Gonzalez, G.; and Others. 2008. BANNER: an executable survey of advances in biomedical named entity recognition. In *Pacific symposium on biocomputing*, volume 13, 652–663. Big Island, Hawaii.

Ling, X., and Weld, D. S. 2012. Fine-Grained Entity Recognition. In *AAAI*.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *Computer Science*.

Nadeau, D. 2007. *Semi-supervised named entity recognition: learning to recognize 100 entity types with little supervision*. Ph.D. Dissertation, University of Ottawa.

Paulheim, H., and Fümkranz, J. 2012. Unsupervised generation of data mining features from linked open data. In *Proceedings of the 2nd international conference on web intelligence, mining and semantics*, 31. ACM.

Ritter, A.; Soderland, S.; and Etzioni, O. 2009. What is this, anyway: Automatic hypernym discovery. In *AAAI Spring Symposium: Learning by Reading and Learning to Read*, 88–93.

Sang, E. T. K., and Hofmann, K. 2009. Lexical patterns or dependency patterns: which is better for hypernym extraction? *International Conference On Computational Linguistics* 8.

Seitner, J.; Bizer, C.; Eckert, K.; Faralli, S.; Meusel, R.; Paulheim, H.; and Ponzetto, S. P. 2016. A large database of hypernymy relations extracted from the web. In *LREC*.

Snow, R.; Jurafsky, D.; and Ng, A. Y. 2004. Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems 17* 17:1297–1304.

Wang, W., and Chang, B. 2016. Graph-based dependency parsing with bidirectional lstm. In *ACL (1)*.

Zhu, X., and Wu, X. 2004. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review* 22(3):177–210.

Publication 7

EMR-based medical knowledge representation and inference via Markov random fields and distributed representation learning

Chao Zhao, Jingchi Jiang, and Yi Guan

EMR-based medical knowledge representation and inference via Markov random fields and distributed representation learning

Chao Zhao ZHAOCHAOCS@GMAIL.COM
Jingchi Jiang JIANGJINGCHI0118@163.COM
Yi Guan* GUANYI@HIT.EDU.CN
School of Computer Science and Technology
Harbin Institute of Technology
Harbin, Heilongjiang, 150001, CHN

Abstract

Objective: Electronic medical records (EMRs) contain an amount of medical knowledge which can be used for clinical decision support (CDS). Our objective is a general system that can extract and represent these knowledge contained in EMRs to support three CDS tasks: test recommendation, initial diagnosis, and treatment plan recommendation, with the given condition of one patient.

Methods: We extracted four kinds of medical entities from records and constructed an EMR-based medical knowledge network (EMKN), in which nodes are entities and edges reflect their co-occurrence in a single record. Three bipartite subgraphs (bi-graphs) were extracted from the EMKN to support each task. One part of the bi-graph was the given condition (e.g., symptoms), and the other was the condition to be inferred (e.g., diseases). Each bi-graph was regarded as a Markov random field to support the inference. Three lazy energy functions and one parameter-based energy function were proposed, as well as two knowledge representation learning-based energy functions, which can provide a distributed representation of medical entities. Three measures were utilized for performance evaluation.

Results: On the initial diagnosis task, 80.11% of the test records identified at least one correct disease from top 10 candidates. Test and treatment recommendation results were 87.88% and 92.55%, respectively. These results altogether indicate that the proposed system outperformed

*. corresponding author

the baseline methods. The distributed representation of medical entities does reflect similarity relationships in regards to knowledge level.

Conclusion: Combining EMKN and MRF is an effective approach for general medical knowledge representation and inference. Different tasks, however, require designing their energy functions individually.

Keywords: Electronic medical record, clinical decision support, medical knowledge network, Markov random fields, distributed representation

1. Introduction

Clinical decision support systems (CDSS) aim to provide clinicians or patients with computer-generated clinical knowledge and patient-related information that can be intelligently filtered or presented at appropriate times, to enhance patient care[1]. A core component of CDSS is the knowledge base, which was once established and updated manually by clinical experts; but is trying to be generated and managed automatically nowadays. This often includes natural language processing (NLP) techniques for mining clinical knowledge to drive CDSS from medical free-text[2], such as medical literature and electronic medical records (EMRs). Focused on the former source, Text REtrieval Conference (TREC) CDS track expected to develop a retrieval-based system to solve three following problems by returning the most relevant biomedical articles [3]:

- Determining a patient's most likely diagnosis given a list of symptoms
- Deciding on the most effective treatment plan for a patient with a known condition
- Determining if a particular test is indicated for a given situation

We would assert, however, that it is possible to shrink the information granularity from articles to medical entities. According to the patient condition, the CDSS can directly provide the proper investigations, diagnosis results, and ordered treatment plans, rather than simply the relevant literatures from which the clinician must draw the necessary information. The EMR is a credible source of

medical knowledge for this purpose - it is the storage of all of a given patient's health care data and medical history of a patient in an electronic format. These data include abundant medical entities, such as the current clinical diagnosis, medical history, results of investigations, treatment plans, and so on[4, 5]. These entities, and relationships between entities, are the primary carriers of medical knowledge in EMR, and can be extracted by the information extraction technique[6, 7, 8]. It shows the possibility of acquiring and organizing medical knowledge automatically based on the EMR. After the information is extracted, the two subsequent key problems are (1) representing medical knowledge via these entities and entity relationships, and (2) making medical inferences according to this representation.

Several machine learning based solutions have been proposed to the above two problems[9], such as the statistical classifiers, association rules, Bayesian networks and so on. Advancements in representation learning and deep learning on NLP have also provided a new approach to CDS. Most methods focus only on one specific disease, however, due to limitations inherent to the model itself or the computational complexity, and universal support systems for general practice remain elusive. We believe it prudent to construct such a general system for two main reasons. First, this kind of system can respond to the demands of ordinary people suffering any problematic symptom. They may want to research independently before seeing a doctor. Second, the general system can yield initial results that may better support real applications than specialized CDS systems. The latter require much more patient information beyond just his or her symptoms and test results to guarantee precise results.

In this paper, we make a preliminary attempt to represent medical knowledge from EMR, to resolve the three problems proposed at the beginning of the paper in a medical entity level. We first represent the medical knowledge using an EMR-based medical knowledge network (EMKN), and then regard it as a Markov random field (MRF) for inference tasks. In the EMKN, nodes are medical entities and edges are entity co-occurrence relationships. The MRF describes the probability

distribution among these entities and makes probabilistic inferences according to the pre-defined energy functions.

Our main contributions are three-fold:

- We proposed a universal EMR-based clinical decision support method using EMKN and MRF. This method takes only the corresponding medical entities as inputs and is not restricted to certain diseases.
- We derived a learning algorithm for arbitrarily derivable energy functions, and integrated the knowledge representation learning approaches into the MRF, to obtain a distributed representation of medical entities.
- We applied the inference architecture to three CDS tasks: test suggestion, initial diagnosis and treatment plan suggestion. This allow us to experimentally demonstrate the efficiency of the proposed method on actual clinical records.

The remainder of this paper is structured as follows. In Section 2, we give a brief review of existing CDS systems as well as the related works of representation learning. In Section 3, we describe the details of construction of EMKN, as well as the inference and learning algorithms based on MRF. Section 4 introduces two distributed representation methods of medical entities to MRF. We evaluated those methods using actual records, as described in Section 5; the results are discussed at length in Section 6. A brief conclusion and discussion on future research directions are presented in Section 7.

2. Related Works

The three problems referred to in Section 1 can be further generalized as one problem: to provide the best possible clinical recommendations (medical investigations, possible diagnosis, and treatment plans) for a given patient's condition. This section introduces previous works relevant to medical knowledge representation and decision making in regards to this problem.

Many of the existing CDS systems focus on one (or one kind of) disease, and adopt classification strategies to solve this problem. Some typical patient features (e.g. signs, symptoms, test results) are extracted with the help of domain expert knowledge, and then transformed and selected. After the feature engineering, the disease condition can be determined by general classifiers like the logistic regression[10, 11], neural network[12, 13] or naïve Bayes classifier[14]. For example, [10] constructed a series of classifiers to predict the estimated glomerular filtration rate (eGFR) of kidney transplant patients, with the help of 56 selected features from the donor and recipient.

Other researchers have attempted to develop models without requiring the manual input of prior knowledge and to depict the relationships among clinical events directly from the data. Association rules mining is a typical approach to identifying relationships of clinical events pairs [15, 16, 17]. Bayesian networks (or "probabilistic graphical models", more generally) can also be used to represent the relationships of medical events[18, 19, 20]. For example, [20] utilized a Bayesian network to implement an adaptive recommendation system to recommend a next order of treatment menu, based on the previous orders. Compared to association rules mining, Bayesian networks can properly account for transitive associations and co-varying relationships among variables.

In attempting to diagnose more than one disease via the approaches described above, the size of the feature set and the number of random variables would be excessive, and neither binary classifiers or Bayesian networks is a good choice. The former would suffer from curse of dimensionality and class imbalance, and the latter are limited by the computational complexity of inference and learning[21]. Non-classification based models are more appropriate. For example, [22] developed a non-disease-specific AI simulation framework via Markov decision process to evaluate the consequences of specific treatment plans; [23] analyzed clinical pathways from clinical workflow log using process mining approaches.

Recent development in representation learning and deep learning have opened new opportunities for medical knowledge representation. Representation learning aims to learn a good representation of the data, which can make it easier to extract useful information when building classifiers[24]. One

widely used representation technique is deep learning[25, 26], which has been particularly successful in a variety of artificial intelligence (AI) fields[27, 28], including NLP[29, 30, 31], where researchers attempted to map the word w to a low-dimensional, dense vector $\boldsymbol{w} \in \mathbb{R}^n$. Different entries of the vector depict the word's features from various aspects. This so-called *distributed representation* method can mitigate data sparsity and improve the generalization power of the model to which it is applied. CBOW and skip-gram[31, 32] are two popular algorithms to obtain such representations.

In medical text processing, researchers have attempted to learn the distributed representation of medical terms using similar approaches. Several have fed unstructured medical copora directly to word2vec toolkits[33], but it is more common to extract the medical concepts from raw text first, and then to learn the representation over the temporal medical concept sequences[34, 35]. The obtained medical concept embeddings can be further applied to the relation extraction[36], patient intention detection[37], and even diagnosis and risk prediction[38, 35, 39]. [35], for example, modeled temporal relations among medical events using recurrent neural networks to efficiently detect heart failure onset. Though the final layer of their model was still a classifier, it only required the clinical events as inputs and all the features used for classification were learned automatically.

Representation learning can also be applied to knowledge representation[40, 41, 42, 43]. Part of human knowledge can be represented in the form of a relation triple (e_h, r, e_t), where there is a certain relationship r from the head entity e_h to the tail entity e_t. Knowledge representation learning (KRL) is deployed to obtain low-dimensional embeddings for entities and relationships of these triples. Typical KRL models include latent factor models (LFMs)[42] and translating embedding (TransE) models [43]. TransE is especially popular due to its prediction accuracy and computational efficiency. To the best of our knowledge, however, these methods have not been applied to learning medical concept embeddings.

3. Methods

The network is a convenient tool for modeling and visualizing entities with complex relationships. In this work, we began by organizing a series of medical entities into a network.

3.1 EMR-based Medical Knowledge Network

We proposed the EMKN, an EMR-based medical knowledge network for knowledge representation from EMR, in [44]. This section gives a brief review and supplementary information about this network.

The corpus we used contained 992 de-identified clinical records[45], which were retrieved from The Second Affiliated Hospital of Harbin Medical University. We manually extracted the medical entities and its modifiers[1]. Medical entities were roughly split into five categories: *symptom*, *test*, *test result*, *disease* and *treatment*. The modifiers included *present*, *possible*, *absent* and the other four modifiers. Based on these entities, we constructed EMKN, where nodes served as medical entities and edges were co-occurrence relationships among entities in one single record.

In this work, We extract three bigraphs from EMKN as listed in Table 1, to support the three problems named in section 1. We denote the bigraph by $G = (\boldsymbol{X}, \boldsymbol{Y})$, where $\boldsymbol{X}$ is the entity set we have observed and $\boldsymbol{Y}$ is our corresponding recommendation items. The diagnosis task, for example, depends only on the SD-EMKN. $\boldsymbol{X}$ denotes the symptom and test result entities, and $\boldsymbol{Y}$ denotes the disease entities. Figure 1 is a visualization of the ENK with its three bigraphs.

1. Although an information extraction system has been developed with the help of these annotated data and a larger database is available, we still used the manually annotated data to eliminate any interferences with the automatic results.

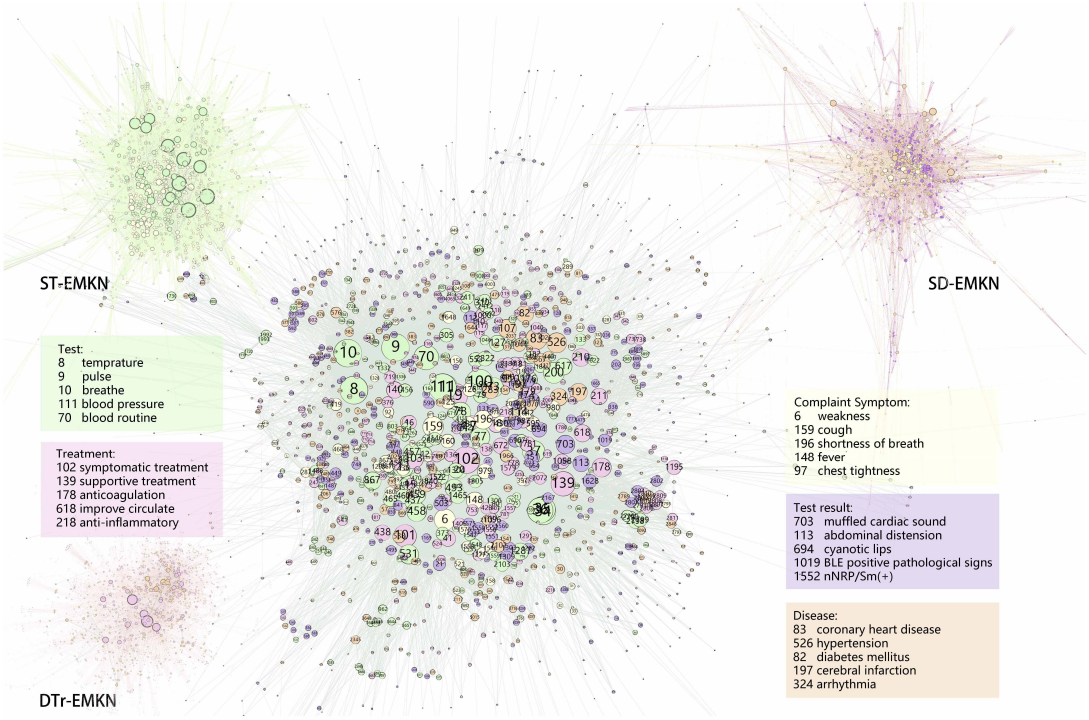


Figure 1: EMKN visualized in Gephi. Nodes sizes are proportional to their degree, and node type is indicated by color. Each node is labeled with a corresponding digital id. Corresponding medical entities are provided for several nodes.

Table 1: Statistical quantities of three EMKN bigraphs. We list the node size of each part, as well as the mean and median of the node degree.

subgraph name	$\boldsymbol{X}$				$\boldsymbol{Y}$			
	type	size	degree mean	degree median	type	size	degree mean	degree median
SD-EMKN	symptom test result	2148	6.85	4	disease	1208	12.19	8
DTr-EMKN	disease	667	8.25	6	treatment	263	10.46	5
ST-EMKN	symptom	811	10.2	8	test	538	15.37	5

3.2 From EMKN to MRF

MRF defines the joint probability among variables $\boldsymbol{A} = (A_1, A_2, \cdots, A_n)$ in terms of an undirected graph. Formally, the joint probability of $\boldsymbol{A}$ can be written as

$$P(\boldsymbol{A}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \phi_C(\boldsymbol{A}_C), \tag{1}$$

where $\mathcal{C}$ is the set of maximal cliques in the graph and $\phi_C(\boldsymbol{A}_C)$ is the corresponding potential of clique C. The potential function $\phi_C : \boldsymbol{A}_C \to \mathbb{R}^+$ defines a map from clique to a positive real number. The larger the value of $\phi_C(\boldsymbol{A}_C = \boldsymbol{a}_C)$, the more likely that $\boldsymbol{A}_C = \boldsymbol{a}_C$. To ensure positivity of $\phi_C(\cdot)$, we rewrote $\phi_C(\cdot) = \exp(-\varepsilon_C(\cdot))$, where $\varepsilon_C(\cdot)$ is called the energy function of the clique C. The smaller the value of $\varepsilon(\boldsymbol{A}_C = \boldsymbol{a}_C)$, the more likely that $\boldsymbol{A}_C = \boldsymbol{a}_C$. Z is the partition function that ensures that $P(\boldsymbol{A})$ follows the probability distribution:

$$Z = \sum_{\boldsymbol{A}} \prod_{C \in \mathcal{C}} \phi_C(\boldsymbol{A}_C) \tag{2}$$

EMKN can be transformed to MRF immediately if its nodes are regarded as random variables. We still use $\boldsymbol{X} = \{X_1, X_2, \cdots, X_m\}$ and $\boldsymbol{Y} = \{Y_1, Y_2, \cdots, Y_n\}$ to denote these variables of $G = (\boldsymbol{X}, \boldsymbol{Y})$. All of Y_i and X_j can take real values from -1 to 1, which indicates the degree of these entities on one patient: we can assign the entities with positive assertion as 1, negative as -1, and others modifiers as 0.5. Entities which do not appeared are set as 0. Inspired by the Ising model,

we define the energy function over the nodes Y_i and X_j here as

$$\varepsilon(Y_i = y_i, X_j = x_j) = f(Y_i, X_j) \cdot (y_i \cdot x_j) \tag{3}$$

$f(Y_i, X_j)$ is a function with value independent of the assignment of Y_i and X_j. When $f(Y_i, X_j) < 0$, the model prefers Y_i and X_j taking the same sign. Conversely, $f(Y_i, X_j) > 0$ implies that Y_i and X_j are more likely to have different signs. Without ambiguity, we also call $f(Y_i, X_j)$ as an energy function.

3.3 Inference on MRF

Given a set of observed variables $\boldsymbol{x} = \{x_1, x_2, \cdots, x_m\}$ of one patient, we calculated the probability of $P(Y_i = 1 | \boldsymbol{X} = \boldsymbol{x})$ and ranked $Y_i \in \boldsymbol{Y}$ accordingly as the result. This is a kind of inference task.

We first calculated $P(\boldsymbol{y}|\boldsymbol{x})$ with the probability distribution defined above.

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{P(\boldsymbol{y}, \boldsymbol{x})}{\sum_{\boldsymbol{Y}} P(\boldsymbol{Y}, \boldsymbol{x})} \tag{4}$$

Notice that all the cliques in $G = (\boldsymbol{X}, \boldsymbol{Y})$ are edges, then we get

$$P(\boldsymbol{y}, \boldsymbol{x}) = \frac{1}{Z} \prod_{i,j,<Y_i,X_j>\in E} \phi(y_i, x_j), \tag{5}$$

where E is the set of edges. Plugging Eq. (5) into Eq. (4), we obtain

$$\begin{aligned} P(\boldsymbol{y}|\boldsymbol{x}) &= \prod_i \frac{\prod_{j,<D_i,S_j>\in E} \phi(y_i, x_j)}{\sum_{Y_i} \prod_{j,<Y_i,X_j>\in E} \phi(Y_i, x_j)} \\ &= \prod_i P(y_i|\boldsymbol{x}) \end{aligned} \tag{6}$$

Eq. (6) shows that the value of Y_i are independent of not only any symptoms which are not its neighbor, but also all other $Y_{\setminus i}$. This allow us to calculate the probability of Y_i separately.

$$P(y_i|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})}\phi(y_i, \boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})}\prod_{j=1}^{|\boldsymbol{X}|}\phi(y_i, x_j) = \frac{1}{Z(\boldsymbol{x})}\exp[-\sum_{j=1}^{|\boldsymbol{X}|}\varepsilon(y_i, x_j)] \tag{7}$$

where $Z(\boldsymbol{x}) = \sum_{Y_i}\phi(Y_i, \boldsymbol{x})$ is the partition function.

3.4 Parameter Learning on MRF

Once we transformed the EMKN to MRF and determined the appropriate inference method, the last step was to learn the parameters of the potential function from the training data. For clarity, we introduce the learning process here with $f_\theta(Y_i, X_j) = \theta_{ij}$; it can still be an arbitrary differential function.

Similar to many discriminative models, we learn the parameter $\boldsymbol{\theta}$ by maximizing the likelihood function of training data:

$$\begin{aligned} L(\boldsymbol{\theta}) &= \sum_{k=1}^{K}\sum_{i=1}^{|\boldsymbol{Y}|}\ln P(y_i^{(k)}|\boldsymbol{x}^{(k)}) - \sum_{i=1}^{|\boldsymbol{Y}|}\sum_{j=1}^{|\boldsymbol{X}|}\frac{\theta_{ij}^2}{2\sigma^2} \\ &= \sum_{k=1}^{K}\sum_{i=1}^{|\boldsymbol{Y}|}(\varepsilon(y_i^{(k)}, \boldsymbol{x}^{(k)}) - \ln Z(\boldsymbol{x})) - \sum_{i=1}^{|\boldsymbol{Y}|}\sum_{j=1}^{|\boldsymbol{X}|}\frac{\theta_{ij}^2}{2\sigma^2} \end{aligned} \tag{8}$$

where K is the number of training records. The second term is a Gaussian prior over the parameters $\boldsymbol{\theta}$. Here, we use the stochastic gradient descend (SGD) to optimize $L(\boldsymbol{\theta})$. The log-likelihood of one single instance is

$$l(\boldsymbol{\theta}) = -\sum_{i=1}^{|\boldsymbol{Y}|}\sum_{j=1}^{|\boldsymbol{X}|}\varepsilon(y_i, x_j) - \sum_{i=1}^{|\boldsymbol{Y}|}\ln Z(\boldsymbol{x}) - \frac{1}{K}\sum_{i=1}^{|\boldsymbol{Y}|}\sum_{j=1}^{|\boldsymbol{X}|}\frac{\theta_{ij}^2}{2\sigma^2} \tag{9}$$

The partial derivation of $l(\boldsymbol{\theta})$ with respect to θ_{ij} is

$$\frac{\partial}{\partial\theta_{ij}}l(\boldsymbol{\theta}) = -\frac{\partial}{\partial\theta_{ij}}\varepsilon(y_i, x_j) - \frac{1}{Z(\boldsymbol{x})}\frac{\partial}{\partial\theta_{ij}}Z(\boldsymbol{x}) - \lambda\theta_{ij} \tag{10}$$

where $\lambda = -1/K\sigma^2$. For brevity, we let

$$g(y_i, x_j) = -\frac{\partial}{\partial \theta_{ij}} \varepsilon(y_i, x_j) \tag{11}$$

then

$$\begin{aligned} \frac{\partial}{\partial \theta_{ij}} l(\boldsymbol{\theta}) &= g(y_i, x_j) - \sum_{Y_i} [\frac{\exp(-\sum_j \varepsilon(y_i, x_j))}{Z(X)} \cdot g(y_i, x_j)] - \lambda \theta_{ij} \\ &= g(y_i, x_j) - \sum_{Y_i} [P(y_i|X) \cdot g(y_i, x_j)] - \lambda \theta_{ij} \\ &= g(y_i, x_j) - \boldsymbol{E}_{P(y_i|X)}[g(y_i, x_j)] - \lambda \theta_{ij} \end{aligned} \tag{12}$$

where $\boldsymbol{E}_P[X]$ is the expectation of X under the distribution P. Once we obtain the partial derivative of $l(\boldsymbol{\theta})$, we can update θ_{ij} with a proper learning rate η

$$\theta_{ij} \leftarrow \theta_{ij} + \eta \frac{\partial}{\partial \theta_{ij}} l(\theta) \tag{13}$$

It is ostensibly necessary to calculate all the $|\boldsymbol{Y}|$ Y-type entities to determine the likelihood in Eq. (8), which is cumbersome and time-consuming. To accelerate the training speed, we sampled the negative Y with the same number of the positive Y from the top-k neighbor list of the given positive X, as measured by the energy function $f(Y_i, X_j)$, to increase the sample possibility of negative Y with high confidence to be positive.

4. Distributed Medical Entity Representation

In last section, we represented each medical entity with an individual node, which is not reasonable. For example, "diabetes" is more similar to the "type II diabetes" than "pneumonia"-this similarity should be reflected in the entity representation. KRL methods are designed to capture this similarity to some degree, by embedding the entities to a low-dimensional, dense vector space.

4.1 Knowledge representation learning

The general idea of KRL is also to construct an energy function for triples. A valid triple has lower energy while an invalid triple has higher energy. The representations of entities and relationships are tuned for this purpose. The LFM model computes the energy of a triple (e_h, r, e_t) by

$$g_{\text{LFM}}(e_h, r, e_t) = -\mathbf{e_h}^T \mathbf{W_r} \mathbf{e_t} \tag{14}$$

where $\mathbf{W_r} \in \mathbb{R}^{d \times d}$ is a transformation matrix and $\mathbf{e_h}, \mathbf{e_t} \in \mathbb{R}^d$ are embeddings of the entities.

The TransE model treats relationships as translations between two entities. For a valid triple (e_h, r, e_t), it hopes that the embeddings satisfy $\mathbf{e_h} + \mathbf{r} \approx \mathbf{e_t}$. The energy function is the distance of two vectors:

$$g_{\text{TransE}}(e_h, r, e_t) = |\mathbf{e_h} + \mathbf{r} - \mathbf{e_t}|_{L_1/L_2} \tag{15}$$

4.2 Medical knowledge representation

Inspired by the above two knowledge representation models, we introduced the distributed representation of medical concepts to the MRF inference architecture.

Here, we use $\mathbf{y_i}, \mathbf{x_j} \in \mathbb{R}^d$ to denote the embeddings of Y_i and X_j, and define the LFM and the Trans model over this pair as

$$f_{\text{LFM}}(Y_i, X_j) = -\text{norm}(\mathbf{y_i}^T \mathbf{W_{xy}}) \mathbf{x_j} \tag{16}$$

$$f_{\text{Trans}}(Y_i, X_j) == |\mathbf{y_i} + \mathbf{r_{xy}} - \mathbf{x_j}|_{L_2} + \gamma \tag{17}$$

$\text{norm}(\mathbf{x}) = \mathbf{x}/|\mathbf{x}|$, $\mathbf{W_{xy}} \in \mathbb{R}^{d \times d}$ is the transformation matrix, and $\mathbf{r_{xy}} \in \mathbb{R}^d$ is the relation embedding. γ is a constant bias which ensures that the f_{Trans} can be negative. We arbitrarily set $d = 100$ and $\gamma = -1$. To alleviate overfitting, we constrained the norm of each entity embedding as 1 after each update.

The parameter learning process is similar to that described in Section 3.4 once we obtained $g(y_i, x_j)$ according to Eq. (11). This learning process is more difficult, however, because the energy functions are coupled together. Once one entity embedding is updated, all the energy values related to this entity change. In Section 3.4, where $f_\theta(y_i, x_j) = \theta_{ij}$, other energy values remain unchanged despite of the update of θ_{ij}.

5. Experiments and evaluation

5.1 Experiments Setup

We randomly selected 700 records as a training set, which was used for EMKN re-construction and parameter learning, and reserved the remaining 292 records as a test set. Experiments were run to evaluate the inference and learning capacity of the MRF-based EMKN on three tasks. The corresponding subgraph and training and test data statistics for each task are listed in Table 2. For each test record, we took the $\boldsymbol{X}$ in subgraph as an input to predict the possibility of $Y_i = 1$ for each $Y_i \in \boldsymbol{Y}$, then re-ranked $\boldsymbol{Y}$ accordingly as result. The golden-standard assigned entities in $\boldsymbol{Y}$ with non-negative modifiers as 1, and others as 0.

Table 2: Training and test data statistics for three tasks.

task	subgraph	training data			test data		
		data size	x per record	y per record	data size	x per record	y per record
symptom/testresult→disease	SD-EMKN	660	6.49	3.79	186	4.9	3.05
disease→treatment	DTr-EMKN	509	4.03	6.89	161	3.36	6.89
symptom→test	ST-EMKN	594	3.96	4.88	165	3.55	3.91

We discarded the training and test records without any positive entity in $\boldsymbol{X}$ or $\boldsymbol{Y}$. New entities also occasionally appeared during the test process, but the knowledge needed is beyond the scope of our EMKN, so we also discarded any test instances with more than half of the new X-type entities.

We used the three energy functions listed above for comparison. In function f_{LFM} and f_{Trans}, we adopted different representations for the same entity in different tasks to explore the potential

ability of models, although the existence of $\mathbf{W}$ and $\mathbf{r}$ allow them to keep consistent. We also designed another three lazy functions as a baseline, which means that they did not have parameters to be learned:

$$\begin{cases} f_{\text{weight}}(Y_i, X_j) = w_{ij} \\ f_{\text{log-w}}(Y_i, X_j) = \log_2(w_{ij} + 1) \\ f_{\text{TF-IDF}}(Y_i, X_j) = \log_2(w_{ij} + 1) \times \log \dfrac{|\boldsymbol{X}|}{\deg(Y_i)} \end{cases} \tag{18}$$

where w_{ij} is the weight between Y_i and X_j. If the edge does not exist, $w_{ij} = 0$. $\deg(Y_i)$ is the degree of Y_i. $f_{\text{log-w}}$ add a log-linear penalty for f_{weight} and $f_{\text{TF-IDF}}$ consider the degree of Y_i further, inspired by TF-IDF: The more neighbors a Y entity has, the weaker that its relationship with each neighbor.

The baseline methods above still use the MRF inference architecture, so we implemented another three baseline models using naïve Bayes, neural networks, and logistic regression. For each record, we regarded $\boldsymbol{X}$ as a feature set, and represented it using a sparse vector. Then we trained individual binary classifier for each $Y \in \boldsymbol{Y}$. Utilizing these methods directly in this way yields very poor results due to the high feature dimension and the class imbalance, so we applied two pre-processing steps. We first sampled the same number of negative instances as the positive instances for each Y. The negative instances which had positive features overlapping with the positive instances were preferred. We then removed the features that were 0 for all the selected training instances to reduce the feature dimension.

5.2 Evaluation measures

Diagnostic support systems for specific diseases have many standard evaluation measures, like ROC curve or AUC, which can not be applied to our evaluations directly. Returning the most probable medical entities from a fixed entity set is more akin to an information retrieval (IR) task. We instead used P@k, R@k, and average precision (AP) to evaluate the performance on single test instance.

P@k defines the fraction of true positive Y entities

$$P@k = \frac{\#(\text{true positive Ys returned in top-k items})}{k} \tag{19}$$

This measure is meaningless for an arbitrary $k(k = 10$, for example). When there is only one positive Y in the test record, its P@10 can no longer more than 0.1. Therefore, we would assign k as the exact number of positive diseases in the evaluated test record.

R@k defines the fraction of relevant Ys that are returned in the top-k items

$$R@k = \frac{\#(\text{true positive Ys returned in top-k items})}{\#(\text{positive diseases in records})} \tag{20}$$

R@k is not a standard evaluation measure in IR because the denominator, which was easy to obtain in our experiment, is hard to estimate in the real retrieval pool. We set $k = 10$ during the evaluation.

AP is the average precision value at the entity list after each true positive entity is returned. That is, if the number of positive Y is m, and we return n of them, ranked as $r_1, r_2, \cdots, r_n$, respectively, then the AP is given by

$$AP = \frac{1}{m} \sum_{i=1}^{n} \frac{i}{r_i} \tag{21}$$

The most ideal condition is that the m results are all returned and ranked at the top of the list, then $AP = 1$.

We also used Mean P@k(MP@k), mean R@k(MR@k), and mean average precision(MAP) measures to evaluate over the whole test set. They are the mean values of the three above measures among all the test instances $\boldsymbol{Q}$:

$$\text{MP@k}(\boldsymbol{Q}) = \frac{1}{|\boldsymbol{Q}|} \sum_{j=1}^{|\boldsymbol{Q}|} P@k(Q_j)$$

$$\text{MR@k}(\boldsymbol{Q}) = \frac{1}{|\boldsymbol{Q}|} \sum_{j=1}^{|\boldsymbol{Q}|} R@k(Q_j)$$

$$\text{MAP}(\boldsymbol{Q}) = \frac{1}{|\boldsymbol{Q}|}\sum_{j=1}^{|\boldsymbol{Q}|} AP(Q_j)$$

6. Results and Discussion

6.1 Evaluation results

The mean evaluation measures of different methods and energy functions for three tasks are listed in Table 3. We prefer to use the R@10 as the primary evaluation measure, so the percentages of test instances with R@10 above 0.1 and 0.9 are also listed in this table. The distribution of these measures over the whole test set is shown in Figure 2.

In initial diagnosis task, there were 80.11% of the test records with R@10 above 0.1, indicating that these records returned at least 1 records in the top-10 results. In test and treatment recommendation tasks, this percentage was 87.88% and 92.55%.

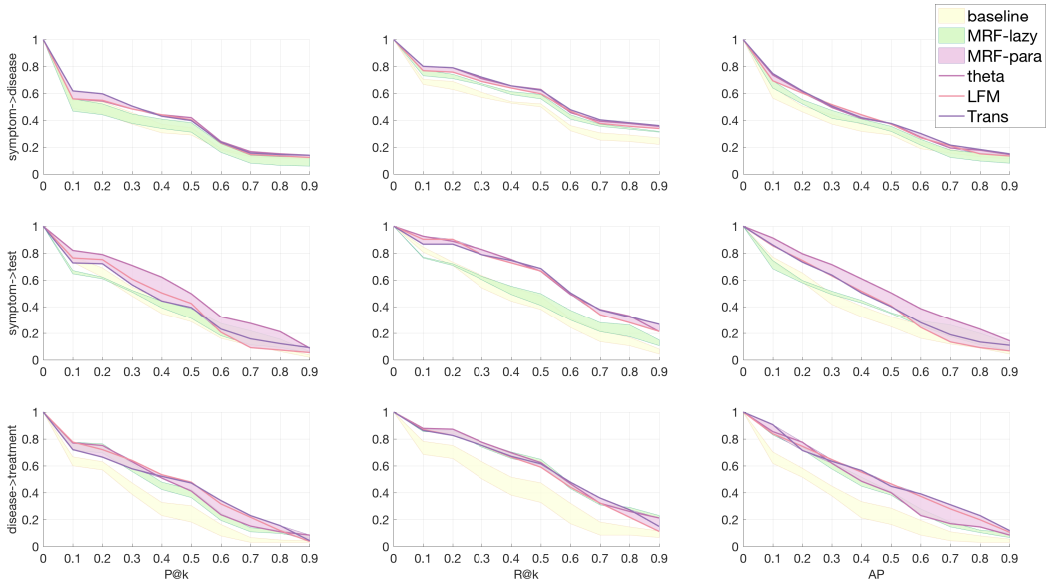


Figure 2: The distribution of measures. Three column show the distribution of P@k, R@k and AP, respectively. Three rows represent the three inference tasks. The x-axis of each subfigure represents the values of each measure, and the y-axis value is the cumulative percentage of records which achieves the performance higher than x-axis value. For clarity, we only draw the results distribution of last three methods in Table 3. The color blocks fill the upper bound and the lower bound of three groups of results: the baseline methods, and the MRF energy functions with or without parameters.

Table 3: Evaluation measures of different methods and energy functions for three tasks.

Methods	Symptom/TestResult→ Disease				
	MP@R	MAP	MR@10	R@10>0.1	R@10>0.9
Naïve Bayes	0.3314	0.3457	0.4707	0.7043	0.2688
Logistic	0.3011	0.3186	0.4468	0.6935	0.2204
Neural network	0.2979	0.3122	0.4537	0.6667	0.2527
Weight	0.2532	0.3191	0.5096	0.7312	0.3118
Log-weight	0.2956	0.351	0.5239	0.7634	0.3172
TF-IDF	0.3224	0.369	0.5321	0.7742	0.3172
Theta	0.3431	0.391	0.5658	**0.8011**	0.3548
LFM	0.3359	0.383	0.5472	0.7688	0.3387
Trans	**0.3543**	**0.4044**	**0.5728**	**0.8011**	**0.3602**

Methods	Symptom→ Test				
	MP@R	MAP	MR@10	R@10>0.1	R@10>0.9
Naïve Bayes	0.3537	0.3598	0.4285	0.8075	0.0745
Logistic	0.3322	0.3212	0.3893	0.8447	0.0435
Neural network	0.3813	0.3709	0.4439	0.8261	0.0807
Weight	0.3234	0.3659	0.4164	0.764	0.1056
Log-weight	0.3438	0.3816	0.4395	0.7702	0.1304
TF-IDF	0.3641	0.4002	0.4606	0.764	0.1491
Theta	**0.4701**	**0.5123**	**0.5939**	**0.9255**	0.2112
LFM	0.3755	0.4175	0.5769	0.903	0.2121
Trans	0.3754	0.4402	0.5894	0.8667	**0.2667**

Methods	Disease→ Treatment				
	MP@R	MAP	MR@10	R@10>0.1	R@10>0.9
Naïve Bayes	0.3057	0.3105	0.4314	0.7576	0.097
Logistic	0.2985	0.2979	0.4189	0.7818	0.1152
Neural network	0.2608	0.2584	0.3416	0.6848	0.0667
Weight	0.3567	0.3979	0.5455	0.8545	0.2242
Log-weight	0.3524	0.4008	0.5422	0.8545	**0.2303**
TF-IDF	0.3969	0.4288	**0.5573**	0.8727	0.2121
Theta	0.3971	0.4258	0.5568	**0.8788**	0.2121
LFM	**0.4161**	0.4669	0.5189	0.8696	0.1118
Trans	0.408	**0.4822**	0.5414	0.8634	0.1491

Generally speaking, the MRF baseline methods outperformed other machine-learning baseline methods. The performance of these baseline methods was enhanced as the energy function complexity increased. After adding parameters to the energy functions, the performance was even further enhanced.

For the diagnostic support system, the Trans model showed optimal performance in all three measures. For the test recommendation task, learning the $\boldsymbol{\theta}$ directly from the data was optimal. For the treatment plan recommendation task, the performance of different energy functions varied across different evaluation measures.

6.2 Medical entity visualization

During the training process of f_{LFM} and f_{Trans}, we obtained the distributed representation of medical entities. These medical embeddings were expected to capture the similarity among entities in regards to knowledge level. To verify this, we reduced the dimension of embeddings from 100 to 2 using the t-Distributed Stochastic Neighbor Embedding (t-SNE) technique for visualization. Figure 3 shows the disease embeddings obtained by Trans model on the diagnosis task. The disease entities are colored in the figure according to the first letter of their ICD-10 code, which indicates their corresponding ICD section. The complete ICD-codes of several disease entities are provided in the enlarged block. It can be seen that similar medical entities indeed stayed close together in the vector space.

We also list several symptom-disease pairs in Figure 4 to illustrate the translation relationships of these entities in vector space. As expected, these relationships are tried to keep parallel to each other.

6.3 Discussion

The three MRF baseline methods use lazy mechanisms and are not equipped with any explicit parameters or learning process. We only need to construct the EMKN from the medical records set, and then calculate the energy values based on the graph measures. These functions are suitable for the online learning of massive flows of data, e.g., for updating the EMKN using daily medical records or other abundant sources of medical knowledge. The performance of these energy functions

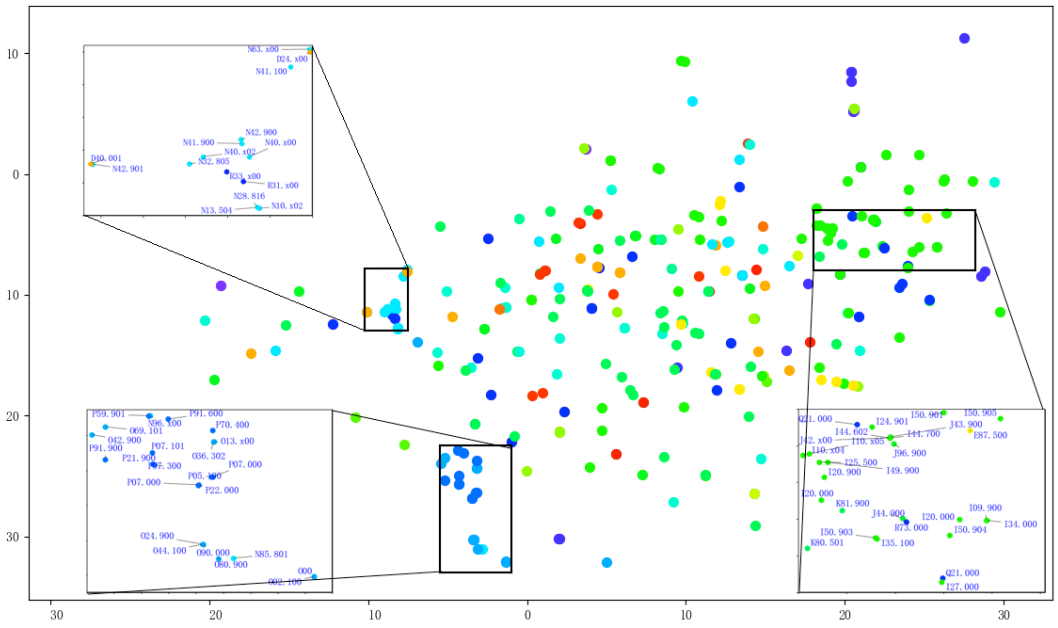


Figure 3: The visualization of disease entities of Trans model.

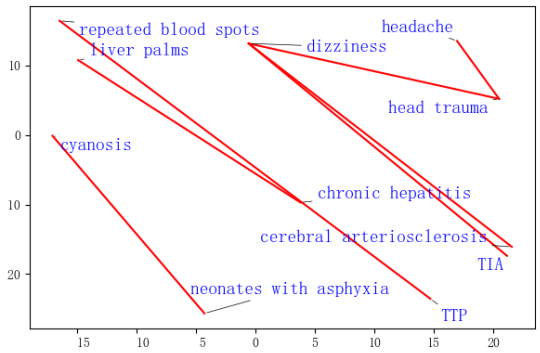


Figure 4: Translation relationships of symptom-disease pairs obtained via Trans model.

increased gradually over the course of our experiment, demonstrating the effectiveness of our method in regards to improving them.

Conversely, the other three MRF energy functions we tested involving parameters, which were learned iteratively from training data. They also performed better than lazy learners. No energy function, however, outperformed others on all three tasks. This differences in performance of the same energy function in different tasks indicates that the energy functions should be designed individually for each task, because each function has its own unique characteristics and application scenarios.

The f_θ sets parameters on the edges of the EMKN, where each edge $< Y_i, X_j >$ has one parameter θ_{ij}. An advantage is that each energy function is independent of the others. The update of θ_{ij} only affects the value of $f(Y_i, X_j)$. However, its parameter size is $O(n^2)$, where n is the number of nodes - this is fairly large compared to our data set.

f_{LFM} and f_{Trans} move the parameters from edges to nodes, by learning the distributed representation of medical entities. This reduces the parameter size to $O(n)$. Smaller parameter size is helpful to reduce the model complexity and alleviate overfitting. This also ensure that entities are no longer independent of each other. Similar entities are close in the vector space, as shown above. Additionally, the representation of entities with low occurrence frequency is affected by the high-frequency entities, which can improve the generalization power of the model on low-frequency knowledge.

The one-to-one relationship assumption is a persistent problem. That is, it is assumed that one head entity is related exactly to one tail entity. These models perform poorly when the real relationships of entities are far from satisfied the assumption. For example, the blood pressure is a common test item for many symptoms with a degree in ST-EMKN is up to 301. As shown in Table 1, the degree median of test entities in ST-EMKN is smaller than that of the diseases in SD-EMKN, but the degree-mean is larger - in effect, there are several test items having extremely large degree. It is hard to learn the embeddings for these test items' symptom neighbors to satisfy

the assumed relationships unless they overlap. For f_θ, however, we only need to increase the weight parameter between the blood pressure and its corresponding symptoms, while other energy functions are not be affected. Therefore, the f_θ performed best on test recommendation task. Another unsatisfactory example of diagnosis task performance is shown in Figure 4. Both head trauma and cerebral arteriosclerosis can cause dizziness, but their embeddings should not be close whatsoever. A better way to alleviate the dissatisfaction of Trans assumption is to subdivide the symptom-disease relationships further, maybe according to the department or the ICD section. Trans and LFM are, after all, multi-relational models.

There is one more limitation to EMKN representation worth noting. The edges between entities represent co-occurrence relationships rather than cause and effect relations, which renders many edges redundant or unnecessary. The existence of these edges makes the model more prone to overfitting.

7. Conclusion

This work is a preliminary attempt to establish general CDSS. We developed a new EMR-driven medical knowledge representation and inference system, with the EMKN, MRF, and representation learning techniques. We used the the current condition of one patient as an input to obtain corresponding recommendations for medical tests, possible diseases, and treatment plans. Six energy functions were proposed and actual clinical records were utilized to evaluate the performance.

The MRF-based inference module outperformed other machine learning baseline methods. The performance was further improved after we introduced the parameters to energy functions. The best system in the diagnosis support task guaranteed that 80.11% of the test records returned at least one right disease out of the top-10 results; these percentages were 87.88% and 92.55% for test and treatment recommendations, respectively. The medical entity embeddings were obtained and evaluated for the expected similarity in knowledge level. None of the methods we tested outperformed

all other methods on all tasks, however, suggesting that the energy function should be individually designed for each task.

In the future, we plan to further refine the entity relationships and the energy functions.

Acknowledgment

We thank the Second Affiliated Hospital of Harbin Medical University for providing the corpus used in this study.

References

[1] M D Osheroff, A Jerome, M D Teich, Jonathan M FHIMSS, M D Levick, M D Saldana, M D Velasco, T Ferdinand, Dean F FHIMSS, M D Rogers, and Others. Improving outcomes with clinical decision support: an implementer's guide. 2012.

[2] Dina Demner-Fushman, Wendy W Chapman, and Clement J McDonald. What can natural language processing do for clinical decision support? *Journal of biomedical informatics*, 42(5):760–772, 2009.

[3] Kirk Roberts, Matthew S Simpson, Ellen M Voorhees, and William R Hersh. Overview of the TREC 2015 Clinical Decision Support Track. In *TREC*, 2015.

[4] TERRY J Hannan. Electronic medical records. *Health informatics: An overview*, pages 133–148, 1996.

[5] David Dagan Feng. *Biomedical information technology*. Academic Press, 2011.

[6] Jon D Patrick, Dung H M Nguyen, Yefeng Wang, and Min Li. A knowledge discovery and reuse pipeline for information extraction in clinical notes. *Journal of the American Medical Informatics Association*, 18(5):574–579, 2011.

[7] Min Jiang, Yukun Chen, Mei Liu, S Trent Rosenbloom, Subramani Mani, Joshua C Denny, and Hua Xu. A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries. *Journal of the American Medical Informatics Association*, 18(5):601, 2011.

[8] Saeed Hassanpour and Curtis P. Langlotz. Information extraction from multi-institutional radiology reports. *Artificial Intelligence in Medicine*, 66:29–39, 2016.

[9] Igor Kononenko. Machine learning for medical diagnosis: History, state of the art and perspective. *Artificial Intelligence in Medicine*, 23(1):89–109, 2001.

[10] Julia Lasserre, Steffen Arnold, Martin Vingron, Petra Reinke, and Carl Hinrichs. Predicting the outcome of renal transplantation. *Journal of the American Medical Informatics Association*, 19(2):255–262, 2012.

[11] Milos Jovanovic, Sandro Radovanovic, Milan Vukicevic, Sven Van Poucke, and Boris Delibasic. Building interpretable predictive models for pediatric hospital readmission using Tree-Lasso logistic regression. *Artificial Intelligence in Medicine*, 72:12–21, 2016.

[12] Daniel Sánchez Morillo, Antonio León Jiménez, and Sonia Astorga Moreno. Computer-aided diagnosis of pneumonia in patients with chronic obstructive pulmonary disease. *Journal of the American Medical Informatics Association*, 20(e1):e111—-e117, 2013.

[13] Filippo Amato, Alberto López, Eladia Mar\'\ia Peña-Méndez, Petr Va\vnhara, Aleš Hampl, and Josef Havel. Artificial neural networks in medical diagnosis. *Journal of applied biomedicine*, 11(2):47–58, 2013.

[14] Wei Wei, Shyam Visweswaran, and Gregory F Cooper. The application of naive Bayes model averaging to predict Alzheimer's disease from genome-wide data. *Journal of the American Medical Informatics Association*, 18(4):370–375, 2011.

[15] Adam Wright, Justine Pang, Joshua C Feblowitz, Francine L Maloney, Allison R Wilcox, Harley Z Ramelson, Louise I Schneider, and David W Bates. A method and knowledge base for automated inference of patient problems from structured data in an electronic medical record. *Journal of the American Medical Informatics Association*, 18(6):859–867, 2011.

[16] A. Wright, A. McCoy, S. Henkin, M. Flaherty, and D. Sittig. Validation of an Association Rule Mining-Based Method to Infer Associations Between Medications and Problems. *Applied Clinical Informatics*, 4(1):100–109, 2013.

[17] X Zhou, S Chen, B Liu, R Zhang, Y Wang, P Li, Y Guo, H Zhang, Z Gao, and X Yan. Development of traditional Chinese medicine clinical data warehouse for medical knowledge discovery and decision support. *Artificial Intelligence in Medicine*, 48(2-3):139–152, 2010.

[18] M Julia Flores, Ann E Nicholson, Andrew Brunskill, Kevin B Korb, and Steven Mascaro. Incorporating expert knowledge when learning Bayesian network structure: a medical case study. *Artificial intelligence in medicine*, 53(3):181–204, 2011.

[19] Marina Velikova, Josien Terwisscha van Scheltinga, Peter J F Lucas, and Marc Spaanderman. Exploiting causal functional relationships in Bayesian network modelling for personalised healthcare. *International Journal of Approximate Reasoning*, 55(1):59–73, 2014.

[20] Jeffrey G Klann, Peter Szolovits, Stephen M Downs, and Gunther Schadow. Decision support from local data: creating adaptive order menus from past clinician behavior. *Journal of biomedical informatics*, 48:84–93, 2014.

[21] David Maxwell Chickering, David Heckerman, and Christopher Meek. Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research*, 5(Oct):1287–1330, 2004.

[22] Casey C. Bennett and Kris Hauser. Artificial intelligence framework for simulating clinical decision-making: A Markov decision process approach. *Artificial Intelligence in Medicine*, 57(1):9–19, 2013.

[23] Zhengxing Huang, Xudong Lu, and Huilong Duan. On mining clinical pathway patterns from medical behaviors. *Artificial Intelligence in Medicine*, 56(1):35–50, 2012.

[24] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[25] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[26] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[27] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, and Others. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[29] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *journal of machine learning research*, 3(Feb):1137–1155, 2003.

[30] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[31] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and hrases and their compositionality. In *NIPS*, pages 1–9, 2013.

[33] Jose Antonio Minarro-Gimenez, Oscar Marin-Alonso, and Matthias Samwald. Exploring the Application of Deep Learning Techniques on Medical Text Corpora. In *Studies in Health Technology and Informatics*, volume 205, pages 584–588, 2014.

[34] De Vine L.a, Zuccon G.b, Koopman B b. C, Sitbon L.a, and Bruza P.b. Medical semantic similarity with a neural language model. *CIKM 2014 - Proceedings of the 2014 ACM International Conference on Information and Knowledge Management*, pages 1819–1822, 2014.

[35] Edward Choi, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association*, 292(3):344–350, 2016.

[36] Xinbo Lv, Yi Guan, Jinfeng Yang, and Jiawei Wu. Clinical Relation Extraction with Deep Learning. *International Journal of Hybrid Information Technology*, 9(7):237–248, 2016.

[37] Chenwei Zhang, Wei Fan, Nan Du, and Philip S Yu. Mining User Intentions from Medical Queries : A Neural Network Based Heterogeneous Jointly Modeling Approach. *Proceedings of the 25th International Conference on World Wide Web*, pages 1373–1383, 2016.

[38] Truyen Tran, Tu Dinh Nguyen, Dinh Phung, and Svetha Venkatesh. Learning vector representation of medical objects via EMR-driven nonnegative restricted Boltzmann machines (eNRBM). *Journal of Biomedical Informatics*, 54:96–105, 2015.

[39] Yu Cheng, Fei Wang, Ping Zhang, Hua Xu, and Jianying Hu. Risk Prediction with Electronic Health Records : A Deep Learning Approach. *SIAM International Conference on Data Mining*, pages 432–440, 2016.

[40] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Conference on Artificial Intelligence*, number EPFL-CONF-192344, 2011.

[41] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

[42] Rodolphe Jenatton, Nicolas L Roux, Antoine Bordes, and Guillaume R Obozinski. A latent factor model for highly multi-relational data. In *Advances in Neural Information Processing Systems*, pages 3167–3175, 2012.

[43] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013.

[44] C. Zhao, J. Jiang, Z. Xu, and Y. Guan. A study of EMR-based medical knowledge network and its applications. *Computer Methods and Programs in Biomedicine*, 143:13–23, 2017.

[45] Bin He, Bin Dong, Yi Guan, Jinfeng Yang, Zhipeng Jiang, Qiubin Yu, Jianyi Cheng, and Chunyan Qu. Building a comprehensive syntactic and semantic corpus of Chinese clinical texts. *Journal of Biomedical Informatics*, 69:203–217, 2017.

Publication 8

Max-margin weight learning for medical knowledge network

Jingchi Jiang, Jing Xie, Chao Zhao, Jia su, Yi Guan, and Qiubin Yu

Max-Margin Weight Learning for Medical Knowledge Network

Jingchi Jiang[a], Jing Xie[a], Chao Zhao[a], Jia Su[a], Yi Guan[a,*], Qiubin Yu[b]

[a]School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China

[b]Medical Record Room, The 2nd Affiliated Hospital of Harbin Medical University, Harbin 150086, China

***Correspondence address: Yi Guan, School of Computer Science and Technology, Harbin Institute of Technology, Comprehensive Building 803, Harbin Institute of Technology, Harbin 150001, China. Tel.: +86-186-8674-8550.**

E-mail addresses: **guanyi@hit.edu.cn (Y. Guan), jiangjingchi@stu.hit.edu.cn (J.C. Jiang), xiejing.hit@gmail.com (J. Xie), hitsa.zc@gmail.com (C. Zhao), sujiahit@gmail.com (J. Su), yuqiubin6695@163.com (Q.B. Yu).**

Abstract

Background and Objective: We propose a topological structure of a medical knowledge network (MKN) for representing first-order knowledge, and we combine it with probabilistic graphical models for medical diagnosis. Compared with a Markov logic network (MLN), MKN not only inherits the expressive representation of MLN but also is more suitable for multivariate diagnosis with numerical symptoms.

Methods: In this paper, we investigate a discriminative weight learning method of MKN based on a max-margin framework. An effective training process is strictly derived for calculating the weight of medical knowledge. The maximum margin medical knowledge network (M^3KN) not only incorporates the inference ability of MKN but also deals with high-dimensional logic knowledge efficiently.

Results: The experimental results indicate that M^3KN achieves a higher F-measure score (73.1%) than the maximum likelihood learning algorithm of MLN. Furthermore, the proposed approach is obviously superior to some classical machine learning algorithms for medical diagnosis. To adequately manifest the importance of domain knowledge, we numerically verify that the diagnostic accuracy of M^3KN is gradually improved with an increase in the number of learned electronic medical records (EMRs), which contain significant medical knowledge.

Conclusions: The results shows that the proposed learning method is better than existing approaches. Our approach can be used for medical diagnosis, and M^3KN can facilitate the investigation of intelligent healthcare.

Keywords*:* Markov logic network; Medical knowledge network; Weight learning; Electronic medical records

1. Introduction

The growth of the aging population and changes in health concepts are widening the gap between healthcare needs and healthcare resources globally. Therefore, providing an effective auxiliary means for assisting clinicians in making efficient clinical decisions has emerged as a critical research topic. Currently, intelligent medical diagnosis [1-3] is recognized as a feasible method for reducing clinician workload and increasing clinician efficiency.

Many existing methods for medical diagnosis involve machine learning algorithms [4] and statistical learning algorithms [5], the most common of which are supervised classifiers, such as support vector machines (SVMs) [6]. By maximizing the margin of confidence, SVMs can deal with high-dimensional features. However, these feature-based algorithms ignore the importance of structure, assigning features independently to each object. In the real world, an inevitable relationship exists between every pair of features. In particular, in the field of medicine, the occurrence of a disease is the consequence of the combined contributions of several inter-related factors. Thus, there is a need for a graphical structured methodology[7] that emphasizes the importance of relationships among different network nodes. Probabilistic graphical models (PGMs) [8-11], which support accurate inference based on relational structured data, are concerned with the application of probabilistic knowledge. As a uniform framework of statistical relational learning, a Markov logic network (MLN) [12]

combines first-order logic with a probability graph model for solving problems of complexity and uncertainty. MLN has been successfully applied to many challenging problems in natural language processing, such as event extraction [13, 14], entity alignment [15, 16], and question-answering systems [17, 18].

The training algorithm is an important component of MLN for calculating the weights of first-order logic knowledge. The initial MLN adopts a Monte Carlo maximum likelihood estimator (MC-MLE) to train the weights with 10 Gibbs chains. However, the convergence speed of MC-MLE is not satisfactory in most cases. In contrast to MC-MLE, some approximation algorithms [19-22] have exhibited empirical success in many applications. Once the initial clause and ground atom satisfiability counts are complete, the knowledge weights can be rapidly calculated via pseudo-likelihood training. Similarly, max-margin methods are competing approaches for discriminative training. By establishing a reasonable margin, the max-margin Markov network (M^3N) [23] and max-margin Markov logic network (M^3LN) [24, 25] can generally improve the accuracy of reasoning and the timeliness of training.

In this paper, we improve MLN to make it more adaptable to medical diagnosis. By incorporating the energy function into the potential function of a Markov network, a medical knowledge network (MKN) can simultaneously model both binary and numerical indexes of symptoms, thereby avoiding the limitation of predicate dimension in MLN. A margin-based optimization problem is defined on the basis of the probability distribution of MKN in order to learn the weight of medical knowledge. Further, we propose a max-margin medical knowledge network (M^3KN) using a sequential minimal optimization (SMO) algorithm for

solving two-variable sub-problems.

The remainder of this paper is organized as follows. Section 2 introduces the Chinese electronic medical records and probabilistic graphical models, namely MLN and MKN. Section 3 defines the max-margin formulation based on MKN and describes the mathematical derivation of SMO learning of M^3KN. Section 4 discusses the experimental evaluation of the proposed approach. Finally, Section 5 concludes the paper and discusses directions for future work.

2. Medical knowledge and probabilistic graphical models

2.1 Chinese electronic medical records

An electronic medical record (EMR) [26] refers to the systematized collection of patient health information in digital format, including free-form text, symbols, charts, and data. As crucial carriers of recorded medical activity, EMRs contain significant medical knowledge [27-29]. Therefore, we employ EMRs as the knowledge foundation and information source of an intelligent medical diagnosis system. Chinese electronic medical records (CEMRs) have been formulated gradually since the reform of China's health system. The structured CEMRs can facilitate knowledge extraction [30] and knowledge annotation [31]. In this study, we focus on CEMRs, including the discharge summary and progress note. In the annotating process [32], the entities are classified into four categories: disease, symptom, test, and treatment, as shown in Table 1A; only the disease entities and symptom entities are extracted to complete the diagnostic task. In addition, seven assertions are annotated for the symptom entities and disease entities, namely present, absent, not associated with the patient, conditional, possible, historical, and occasional. Table 1B lists the assertions of the medical

entities with examples.

Table 1

Entities and their assertions annotated in CEMRs

Part A

Entity type	Example
Diseases	行支气管镜检查示：小细胞肺癌(Bronchoscopy showed: small cell lung cancer)
Symptoms	疼痛时伴右下肢活动受限(Pain accompanied by the right lower extremity activity limitation)
Tests	行支气管镜检查示：小细胞肺癌(Bronchoscopy showed: small cell lung cancer)
Treatments	注射胰岛素控制血糖(Injection of insulin to control blood glucose)

Part B

Assertion type	Description	Example
Present	Disease or symptom exists in the patient.	头 CT ***示***：双侧多发腔梗(head CT ***showed***: bilateral multiple lacunar infarct)
Absent	Disease or symptom does not exist in the patient.	双下肢***无***浮肿(***no*** edema in both lower limbs)
Possible	Disease or symptom may exist in the patient.	右肺下叶***考虑***创伤性湿肺(Right lung lower lobe ***consider*** traumatic wet lung)
Conditional	Disease or symptom occurs in the patient under certain conditions.	胸闷、气短，常于***饮酒后***出现(chest tightness, shortness of breath, commonly occurs ***after drinking***)
Not associated with the patient	Disease or symptom exists in the patient's relatives.	患者***父母***均患有糖尿病(***parents*** of the patient suffer from diabetes)
Occasional	Disease or symptom exists in the	***时有***胸闷气短(chest tightness and shortness of

	patient occasionally	breath occur ***sometimes***)
Historical	Disease or symptom has existed in the past	***18 年前***患有肺炎(pneumonia ***18 years of age***)

2.2 Markov logic network (MLN)

As a uniform framework of statistical relational learning, a Markov logic network (MLN) combines weighted first-order clauses with a probability graph model for solving problems of complexity and uncertainty. It provides a means for softening first-order logic by making situations in which not all clauses are satisfied less likely but not impossible. An MLN can be understood as the joint distribution of a set of variables x, which can describe a real-world scenario. The probability of a particular truth assignment x to the variables in X is defined as

$$P(X = x) = \frac{1}{Z}\prod_i \phi_i(x_{\{i\}})^{n_i(x)} = \frac{1}{Z}\exp(\sum_i \omega_i n_i(x)) \quad , \tag{1}$$

where $x_{\{i\}}$ is 1 if $x_{\{i\}}$ is satisfied and 0 otherwise, ω_i is the weight associated with clause $f_i \in F$, $n_i(x)$ is the number of groundings of the *i*th clause f_i that are satisfied given the current truth assignment to the variables in X, and Z is the normalization constant.

To apply MLN in medical diagnosis, the atom is considered as the medical entity. When the medical entity presents an explicit condition for a patient, the corresponding atom of this entity in MLN is assigned the value 1; otherwise, it is assigned the value 0. Given a series of symptoms, the risk probability for a specific disease is calculated as

$$\arg\max_y P(y \mid x) = \arg\max_y \sum_i \omega_i n_i(x, y) \quad . \tag{2}$$

The maximum a posteriori probability (MAP) inference in MLN is equivalent to searching for the truth assignment that maximizes the sum of the weights of satisfied clauses.

It is an NP-hard problem for the calculation of $n_i(x, y)$, the general resolution of which is given by MaxWalkSAT [33].

In discriminative learning of weights, some predicates will be activated as evidence and others will be queried. By maximizing the conditional log likelihood (CLL), the weight parameter is adjusted for the purpose of correctly predicting the queries given the evidence.

2.3 Medical knowledge network (MKN)

Although MLN has a complete theoretical framework, it is suitable only for binary predicates. In the field of healthcare, the indexes of symptoms are often expressed in numeric or discrete form. If the symptom and the disease are considered as the evidence and the query, respectively, the existing MLN methodology has some obvious shortcomings for medical diagnosis. In our previous work [34], we addressed this problem. By changing the form of expression of the potential function, we incorporate the continuous variable x into the joint distribution of MLN; thus, the conditional probability model can be deduced by a Boltzmann machine [35].

In a medical knowledge network (MKN), a node and an edge are expressed by a named entity and an entity relationship, respectively, which are extracted manually from EMRs by referencing the medical concept annotation guideline and the assertion annotation guideline given by Informatics for Integrating Biology and the Bedside (i2b2) [36]. The potential function $\phi(D)$ of MKN can be regarded as the state of a clique D, which is composed of one or more entity relationships. Based on statistical physics, the potential function $\phi(D)$ can also be expressed as an energy function $\varepsilon(D)$ [37]. Intuitively, an energy function captures the affinities between interacting entities. More precisely, we can rewrite $\phi(D)$ as

$\phi(D) = \exp(-\varepsilon(D))$, where $\varepsilon(D) = -\ln\phi(D)$ is often called an energy function. One of the earliest types of Markov network (MN) models is the Ising model [38], which first emerged in statistical physics as a model for the energy of a physical system involving a system of interacting atoms. In this study, the energy function associated with an edge is defined by a particularly simple parametric form: $\varepsilon_{i,j}(x_i, x_j) = -\omega_{i,j} x_i x_j$, where x_i and x_j represent the values of the two sides of the edge, namely the "symptom" entity and the "disease" entity, respectively, and $\omega_{i,j}$ is the contribution of the clique D. Based on the formula of probability calculation of MN, the joint distribution of MKN is defined as the product of the potential functions

$$\begin{aligned} P(y \mid x^s, x^d) &= \frac{1}{Z}\prod_{i \atop f_i \in F} \phi_i(D) = \frac{1}{Z}\prod_{i \atop f_i \in F} \exp(-\varepsilon(D)) \\ &= \frac{1}{Z}\exp(-\sum_{i \atop f_i \in F}(-\omega_i x^s_{f_i} x^d_{f_i})) \\ &= \frac{1}{Z}\exp(\sum_{i \atop f_i \in F}\omega_i x^s_{f_i} x^d_{f_i}) \end{aligned} \quad , \tag{3}$$

where F is the set of ground clauses in which the symptom variable $x^s_{f_i}$ appears.

In summary, MKN is an improved algorithm for overcoming the disadvantage of MLN in numeric-based diagnosis. By extracting medical knowledge from CEMRs, a medical knowledge network is constructed; it is composed of "disease" nodes and "symptom" nodes. To address the problem of discrete variables, the energy function defined by the Ising model is incorporated into the potential function. According to Eq. (3), the final risk of a specific disease y will be calculated when the set of clauses F, symptom vector x^s, and disease vector x^d are given.

3. Materials and methods

3.1 Maximum margin function

The objective of a discriminative weight learner for MKN is to optimize the conditional probability $P(y|x)$ by adjusting the weight vector ω of medical knowledge. We replace the optimization problem by a differentiable proportion function:

$$\frac{P(y|x,\omega)}{P(\hat{y}|x,\omega)} \quad , \tag{4}$$

where the numerator is the probability of the correct disease y and the denominator is the probability of the closest incorrect disease $\hat{y} = \arg\max_{\bar{y} \in Y \setminus y} P(\bar{y}|x)$. When the ratio is maximum, the vector ω is considered as the optimal weight vector. Taking the logarithm of Eq. (4), we express the reasonable margin of MKN as

$$\gamma(x, y, \omega) = \omega^T x\, y - \omega^T x\, \hat{y} \quad . \tag{5}$$

The max-margin problem above can be formulated as a margin optimization of structural SVMs. According to the derivation procedures of the maximum margin classifier of SVMs, the objective function $\gamma(x, y, \omega)$ can be transformed into $\min_{\omega} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n} \varepsilon_i$, and the corresponding constraint condition is $\omega^T[x_i y_i - x_i \hat{y}_i] \geq \Delta t - \varepsilon_i;\ \ \varepsilon_i \geq 0, i = 1,...,n$, where ε_i is the slack variable, Δt parameterizes the loss function, and C controls the weight between the hyperplane and the deviation value. By substituting the constraint condition into the objective function, we can define the complete form of the Lagrange function as [39]

$$\begin{aligned} \max_{\alpha_i \geq 0} L(\omega, \varepsilon, \alpha) = \max_{\alpha_i \geq 0} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n} \varepsilon_i - \sum_{i=1}^{n} \alpha_i [\omega^T (x_i y_i - x_i \hat{y}_i) - \Delta t + \varepsilon_i] \\ s.t., \quad 0 \leq \alpha_i \end{aligned} \quad , \tag{6}$$

where α is the Lagrange multiplier. Then, the max-margin problem is equivalent to minimize $\max_{\alpha_i \geq 0} L(\omega, \varepsilon, \alpha)$, which is expressed as $\min_{\omega} \max_{\alpha_i \geq 0} L(\omega, \varepsilon, \alpha)$. However, it is

extremely difficult for the solving process based on the two parameters ω and α to achieve global optimization, especially with regard to the inequality constraints α_i. Therefore, the optimization problem needs to be replaced by a dual problem $\max_{\alpha\geq 0}\min_{\omega} L(\omega,\varepsilon,\alpha)$ that is feasible [40]. In this case, we can indirectly solve the original problem by searching for the optimal solution of the dual problem.

3.2 SMO learning of M3KN

To solve the convex quadratic programming problem (QP) of the dual function, we present a sequential minimal optimization (SMO) algorithm. The dual problem $\max_{\alpha\geq 0}\min_{\omega} L(\omega,\varepsilon,\alpha)$ can be split into two parts: minimize $L(\omega,\varepsilon,\alpha)$ with respect to ω and maximize $\min_{\omega} L(\omega,\varepsilon,\alpha)$ with respect to α. For the former optimization, we adopt the calculation of partial derivatives, which is a traditional and effective method for most extremum questions. By setting $\partial L/\partial\omega$ to zero, we can obtain $\omega=\sum_{i=1}^{n}\alpha_i(x_i y_i - x_i\hat{y}_i)$, where ω is expressed by α. For convenient calculation, we transform $\min_{\omega} L(\omega,\varepsilon,\alpha)$ into $-\max_{\omega} L(\omega,\varepsilon,\alpha)$, which has the following form:

$$\min_{\omega} L = -\max_{\omega} L = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j K_i K_j - \sum_{i=1}^{n}\alpha_i\Delta t \qquad s.t., \quad \sum_{i=1}^{n}\alpha_i = C \;\; ; \;\; 0\leq\alpha_i\leq C, \quad (7)$$

where $x_i y_i - x_i\hat{y}_i$ is rewritten as K_i. Then, let us consider the original dual formulation again. We expand the Lagrange function into the polynomial form of the variable.

$$\begin{aligned}\max_{\alpha}\min_{\omega} L = \frac{1}{2}[&\alpha_1^2K_1^2 + 2\alpha_1\alpha_2K_1K_2 + \alpha_2^2K_2^2 + 2\sum_{i=3}^{n}\alpha_1\alpha_iK_1K_i + 2\sum_{i=3}^{n}\alpha_2\alpha_iK_2K_i \\ &+ \sum_{i=3}^{n}\sum_{j=3}^{n}\alpha_i\alpha_jK_iK_j] - \sum_{i=3}^{n}\alpha_i\Delta t - \alpha_1\Delta t - \alpha_2\Delta t\end{aligned} \quad (8)$$

In general, the SMO approach solves this QP by analytically optimizing two-variable subproblems [41]. The main idea is to first update one pair of Lagrange multipliers, α_1 and

α_2, while all other parameters remain unchanged. Then, the next pair of multipliers will be updated similarly until all multipliers have reasonable values. Before multiplier calculation, Eq. (9), which combines the constraint conditions of Eq. (7), needs to be satisfied.

$$\alpha_1^{old} + \alpha_2^{old} = \alpha_1^{new} + \alpha_2^{new} = \sum_{i=1}^{n} \alpha_i - \sum_{i=3}^{n} \alpha_i = C - \sum_{i=3}^{n} \alpha_i = T \tag{9}$$

By employing α_2 deduced from Eq. (9) instead of α_1, the dual function for margin maximization is rewritten as

$$\begin{aligned} \max_{\alpha} \min_{\omega} L = \frac{1}{2}[(T-\alpha_2)^2 K_1^2 + 2T\alpha_2 K_1 K_2 - 2\alpha_2^2 K_1 K_2 + \alpha_2^2 K_2^2 + 2\sum_{i=3}^{n} T\alpha_i K_1 K_i \\ - 2\sum_{i=3}^{n} \alpha_2 \alpha_i K_1 K_i + 2\sum_{i=3}^{n} \alpha_2 \alpha_i K_2 K_i + \sum_{i=3}^{n}\sum_{j=3}^{n} \alpha_i \alpha_j K_i K_j] - \sum_{i=3}^{n} \alpha_i \Delta t - T\Delta t \end{aligned} . \tag{10}$$

Further, we naturally calculate the partial derivative of $\max_{\alpha} \min_{\omega} L$ with respect to α_2. Meanwhile, we aim to obtain the optimal multiplier α_2 by solving Eq. (10). The mathematical derivation is expressed as follows:

$$\begin{aligned} &\frac{\partial L}{\partial \alpha_2} = \frac{1}{2}[-2(T-\alpha_2)K_1^2 + 2TK_1K_2 - 4\alpha_2 K_1 K_2 + 2\alpha_2 K_2^2 - 2\sum_{i=3}^{n} \alpha_i K_1 \alpha_i + 2\sum_{i=3}^{n} \alpha_i K_2 \alpha_i] = 0 \\ &\Rightarrow -TK_1^2 + \alpha_2 K_1^2 + TK_1K_2 - 2\alpha_2 K_1 K_2 + \alpha_2 K_2^2 - \sum_{i=3}^{n} \alpha_i K_1 K_i + \sum_{i=3}^{n} \alpha_i K_2 K_i = 0 \\ &\Rightarrow \alpha_2 (K_1^2 - 2K_1K_2 + K_2^2) = TK_1^2 - TK_1K_2 + \sum_{i=3}^{n} \alpha_i K_1 K_i - \sum_{i=3}^{n} \alpha_i K_2 K_i \end{aligned} , \tag{11}$$

where T can be expressed as $\alpha_1^{old} + \alpha_2^{old}$ and α_2 is equivalent to α_2^{new}. Therefore, proper function transformation for Eq. (11) makes it possible to accurately calculate α_2^{new}. However, the calculation of $\sum_{i=3}^{n} \alpha_i K_i$ is an intractable problem, requiring operation over the entire sample.

For solving the problem of the finite sum formula, we need to review the definition of Markov network. A Markov network is composed of an undirected graph G and a set of

potential functions. The graph has a node for each variable, and the model has a potential function for each clique in the graph. Another important property of a Markov network is that the potential energy of a clique is equal to the potential product of variables in the clique. Combined with the potential function of MKN, the inferential reasoning process of the finite sum formula can be expressed as

$$\begin{aligned}
&\phi(D)=\phi(x)\phi(y)=\exp(-\varepsilon(D))\\
&\Rightarrow \log\phi(x)\phi(y)=-\varepsilon(D)=\omega xy\\
&\Rightarrow \omega=\frac{\log\phi(x)+\log\phi(y)}{xy}=\sum_{i=1}^{n}\alpha_i K_i \qquad , \qquad (12)\\
&\Rightarrow \sum_{i=3}^{n}\alpha_i K_i=\omega-\alpha_1 K_1-\alpha_2 K_2=\frac{\log\phi(x)+\log\phi(y)}{xy}-\alpha_1 K_1-\alpha_2 K_2
\end{aligned}$$

where $\phi(x)$ and $\phi(y)$ denote the potential energy of variable x and y, respectively. Once $\sum_{i=3}^{n}\alpha_i K_i$ is known, the multiplier α_2 is updated by substituting the finite sum formula into the last equation of Eq. (11).

$$\begin{aligned}
&\alpha_2^{temp}(K_1-K_2)^2=(\alpha_1^{old}+\alpha_2^{old})K_1^2-(\alpha_1^{old}+\alpha_2^{old})K_1K_2+(\frac{\log\phi(x)+\log\phi(y)}{xy}-\alpha_1^{old}K_1-\alpha_2^{old}K_2)K_1\\
&\qquad -(\frac{\log\phi(x)+\log\phi(y)}{xy}-\alpha_1^{old}K_1-\alpha_2^{old}K_2)K_2 \qquad .(13)\\
&\Rightarrow \alpha_2^{temp}(K_1-K_2)^2=\alpha_2^{old}(K_1-K_2)^2+\frac{\log\phi(x)+\log\phi(y)}{xy}(K_1-K_2)\\
&\Rightarrow \alpha_2^{temp}=\alpha_2^{old}+\frac{\log\phi(x)+\log\phi(y)}{xy(K_1-K_2)}
\end{aligned}$$

Considering the restriction conditions of the multiplier, we get the final solution of α_2.

$$\alpha_2^{new}=\begin{cases}L & \alpha_2^{temp}\le L\\ \alpha_2^{temp} & L<\alpha_2^{temp}<H\\ H & \alpha_2^{temp}\ge H\end{cases}, \qquad (14)$$

where L and H denote $\max(0,T-C)$ and $\min(C,T)$, respectively. According to the first equation of Eq. (9), we can calculate α_1 directly under the known α_2 and the initial value of the multiplier. In accordance with the above approach, we update a pair of multipliers each

time, which is known as a two-variable optimization subproblem, until all the multipliers are recalculated. After several iterations, the multipliers will converge gradually to an approximate optimal solution. Finally, by denoting the expression of weight as $\omega = \sum_{i=1}^{n} \alpha_i (x_i y_i - x_i \hat{y}_i)$, it is easy to obtain the optimal knowledge weight.

4. Experiment and discussion

4.1 Corpus

For this study, we adopted Chinese electronic medical records (CEMRs) as the corpus. These CEMRs, from which protected health information (PHI) [42] was removed, were acquired from the Second Affiliated Hospital of Harbin Medical University, and we obtained the usage rights for research. By referencing the medical concept annotation guideline and the assertion annotation guideline given by Informatics for Integrating Biology and the Bedside (i2b2), we drafted the guidelines for CEMRs [32] and manually annotated the named entity and entity relationship of 992 CEMRs as the resource of medical knowledge. In addition, we randomly selected 300 unlabeled CEMRs as the test set and employed the conditional random field (CRF) algorithm to recognize the named entities [43,44], including the symptom entities and disease entities.

4.2 Metrics

We adopted the F-measure to measure the performance of the M^3KN learning algorithm. In a diagnosis task, our method considers 10 disease candidates and sorts the confidence scores in descending order. If one CEMR has *m* definite diseases and M^3KN returns *n* of them, then the R@10 (recall for the first 10 results) is given by

$$R@10 = \frac{n}{m} \qquad 0 \le n \le m, n \le 10 \quad . \tag{15}$$

The mean value of recall is $\bar{R}@10 = \sum_l R@10 / l$, where l denotes the number of test samples. In contrast to the recall, the P@10 (precision for the first 10 results) is defined as the ratio of correctly diagnosed cases, i.e., at least one definite disease is presented in the first 10 results, to the total number of cases:

$$\bar{P}@10 = \frac{r}{l} \qquad r \le l \quad . \tag{16}$$

It measures the diagnostic accuracy of medical knowledge, whereas the recall focuses on the diagnostic coverage. In addition, we used the weighted harmonic mean of recall and precision, namely the F-measure, as a comprehensive evaluation indicator to verify the effectiveness of M^3KN.

4.3 Parameter Selection

Next, a reasonable selection of parameters was considered. In the methodology of M^3KN, there are two uncertain parameters: the weighting parameter C and the potential function ϕ, which directly influence the effectiveness of the learning algorithm. Therefore, two experiments were conducted to choose the optimal parameter C and the appropriate expression ϕ, respectively.

Evaluation of the quality of the parameters is an essential step. In this paper, we adopted diagnostic effects as the common standard to determine whether the selected parameters enable the knowledge weight to facilitate disease diagnosis. First, we set the weighting parameter C to 100 and experimented with three classical measures for the potential function: degree, PageRank, and betweenness centrality. Fig. 1 shows the distribution of R@10 for 992 training samples using the three above-mentioned measures. The horizontal axis represents

the value of R@10 and vertical axis represents the number of CEMRs. The distributions of degree, PageRank, and betweenness centrality are denoted by blue, green, and red bars, respectively. As a result, 549 of the 992 training samples achieved 100% recall when the degree was defined as the potential function. Under the other two measures, more samples were concentrated in a low range of recall. Thus, we believe that degree is more suitable as the potential function.

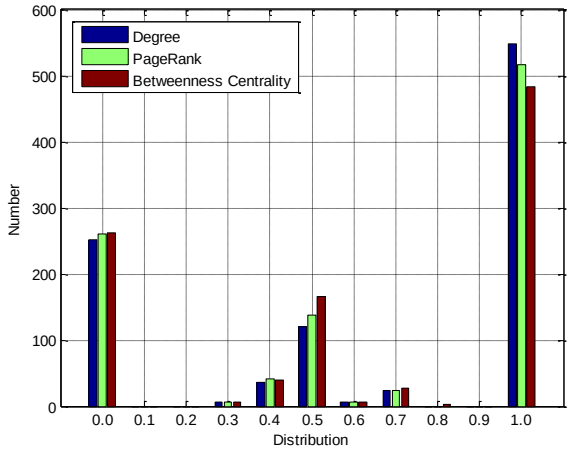


Fig. 1. Distribution of $R@10$ for 992 training samples using different potential measures.

Then, we experimentally selected another parameter C to control the weight between the hyperplane and the deviation value. In the range of 100 to 2500, we uniformly selected 25 data points as C and verified the effectiveness of each value in disease diagnosis. Fig. 2 shows the change in recall, precision, and F-measure under different C. In the initial part of the experiment, the diagnostic effects changed significantly as the value of C increased. When C was close to 1700, all the indicators showed the highest values; subsequently, the indicators remained stable. Therefore, we set the weighting parameter C to 1700.

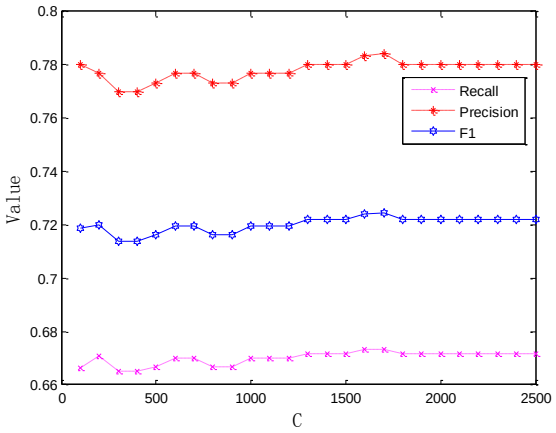


Fig. 2. Selection of weighting parameter *C*.

In addition to the above parameters, owing to the lack of numerical indexes for symptoms and diseases in our CEMRs, we adopted seven assertions as discrete variables to represent the symptom *x* and disease *y*. According to the degree of seriousness, seven assertions were divided into four grades and each grade was assigned a weight, as summarized in Table 2.

Table 2

Assertions and their grades in M3KN

Grade	Assertion type
1	Absent & Not associated with the patient
2	Historical & Possible
3	Occasional & Conditional
4	Present

4.4 Comparison with other algorithms

To verify the effectiveness of M^3KN from the aspects of learning and diagnosis, we compared nine methods, including six types of machine learning approaches, Markov logic

network, MKN based on maximum likelihood [34], and M^3KN. In the diagnosis, the traditional machine learning approach was used for multi-class classification whereby symptoms are extracted as the feature vectors and diseases are considered as the object class. However, the experimental results were not satisfactory. Consequently, we built a two-class processing model for each disease that appears in the 992 CEMRs. In the testing phase, all the two-class models were applied to 300 test CEMRs. For each test CEMR, we calculated the confidence of each disease and ranked the disease candidates by confidence. By selecting the top 10 diseases from the candidate list as the diagnosis result, we could use the above-mentioned metrics to evaluate the validity of the model. In the experiment, we found that the number of positive training samples is far smaller than the number of negative samples, which resulted in reduction of the number of positive samples in testing. Hence, we randomly selected some negative samples from the entire negative set and ensured that the ratio of the number of positive and negative samples was maintained at 0.25. Owing to the randomness of selecting the negative samples, the trained model exhibited fluctuation. To verify the degree of fluctuation, we repeated the experiment 40 times for each model and calculated the variance of the F-measure, as shown in Table 3 and Fig. 3. The results indicated that the fluctuation was controlled in a reasonable range, and it did not affect the assessment of the diagnosis.

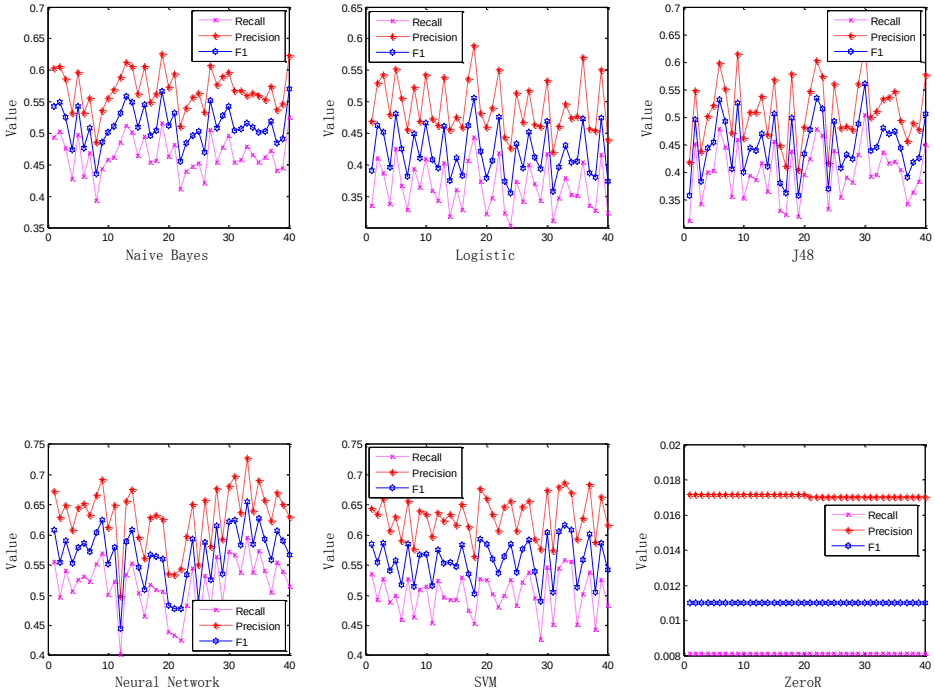


Fig. 3. Degree of fluctuation of six machine learning algorithms.

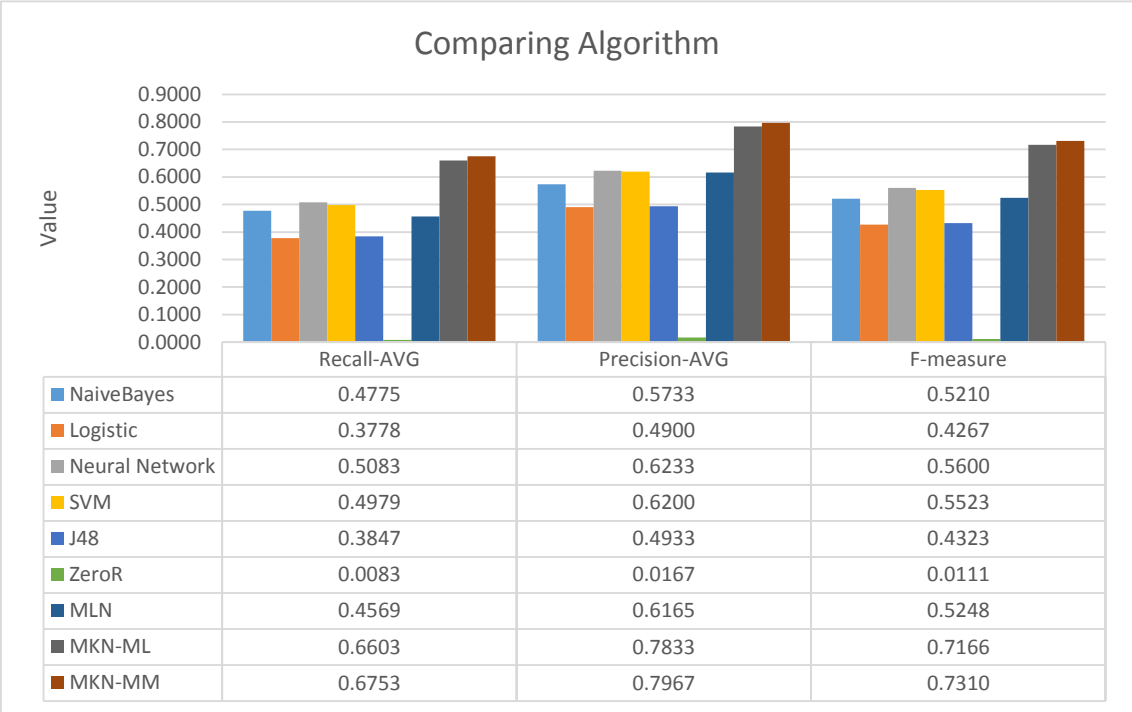
Table 3

Variance of diagnostic effects of the machine learning algorithms.

	Naïve Bayes	Logistic	J48	Neural Network	SVM	ZeroR
Variance	0.8778	0.1475	0.2721	0.2165	0.1082	4.227×10^{-10}

In contrast to machine learning methods, probabilistic graphical models (PGM), such as MLN and MKN, usually apply knowledge to make inferences. In this study, the medical knowledge was extracted in the form of first-order logic, which was transformed from the entity relationships of 992 CEMRs. By integrating the medical knowledge, we constructed a knowledge network consisting of entities as nodes and relationships as edges. Using the knowledge network, we evaluated the impact of MLN, MKN based on maximum likelihood (MKN-ML), and MKN based on maximum margin (MKN-MM) on diagnosis. We employed Tuffy [45] as the base MLN solver and generated the available formalization of knowledge

for Tuffy. The performances of the machine learning methods and probabilistic graphical models are shown in Fig. 4.



	Recall-AVG	Precision-AVG	F-measure
NaiveBayes	0.4775	0.5733	0.5210
Logistic	0.3778	0.4900	0.4267
Neural Network	0.5083	0.6233	0.5600
SVM	0.4979	0.6200	0.5523
J48	0.3847	0.4933	0.4323
ZeroR	0.0083	0.0167	0.0111
MLN	0.4569	0.6165	0.5248
MKN-ML	0.6603	0.7833	0.7166
MKN-MM	0.6753	0.7967	0.7310

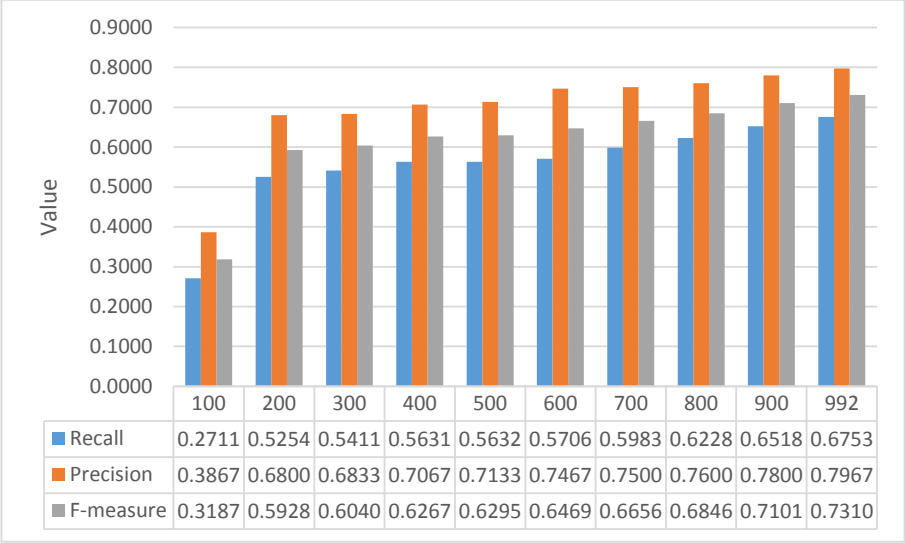
Fig. 4. Comparison of nine diagnostic algorithms.

From Fig. 4, it can be seen that three indicators were used to evaluate the performance of nine diagnosis models. Under the machine learning methods, the F-measure was generally low such that most of them were mainly concentrated between 0.01 and 0.56. In contrast to the machine learning methods, probabilistic graphical models can improve the effectiveness of disease diagnosis significantly. In particular, under MKN-MM, the diagnostic result that the F-measure exceeds 0.73 is much more reliable than that of the other methods. Further analyses showed that the effectiveness of the maximum-margin-based algorithm was obviously superior to that of the maximum-likelihood-based algorithm using the same inference model. In other words, more reasonable knowledge weights that play an important role in disease reasoning can be trained by the maximum margin learning algorithm. Thus, it

can be concluded that the results of diagnosis are significantly improved by MKN-MM, further demonstrating the significance of the maximum margin learning algorithm for the reliability of our medical knowledge.

4.5 Significance of medical knowledge

In this section, we discuss the significance of medical knowledge and investigate the structural changes in the knowledge network. To verify the importance of medical knowledge, we assumed that the diagnostic accuracy could be improved by increasing medical knowledge. In a follow-up experiment, the 992 annotated CEMRs were divided randomly into 10 groups. We constructed the training set in an incremental manner until 10 groups were complete. The max-margin learning algorithm and MKN-based inference algorithm were applied to these training sets. The performance of diagnosis with different numbers of training sets is shown in Fig. 5.

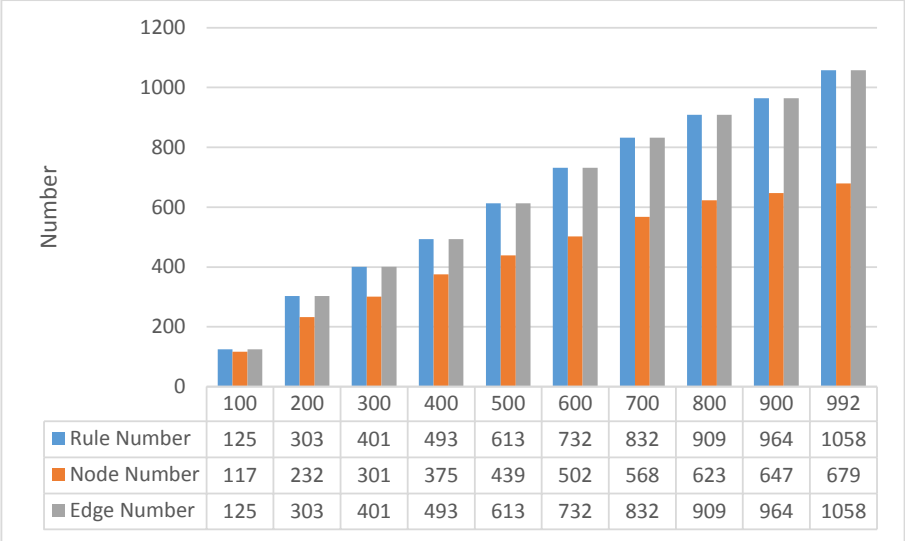


	100	200	300	400	500	600	700	800	900	992
Recall	0.2711	0.5254	0.5411	0.5631	0.5632	0.5706	0.5983	0.6228	0.6518	0.6753
Precision	0.3867	0.6800	0.6833	0.7067	0.7133	0.7467	0.7500	0.7600	0.7800	0.7967
F-measure	0.3187	0.5928	0.6040	0.6267	0.6295	0.6469	0.6656	0.6846	0.7101	0.7310

Fig. 5. Changes in diagnostic effects with increasing number of CEMRs.

It was observed that an increase in the number of CEMRs could obviously enhance both recall and precision. Combined with Fig. 6, these results further prove the importance of

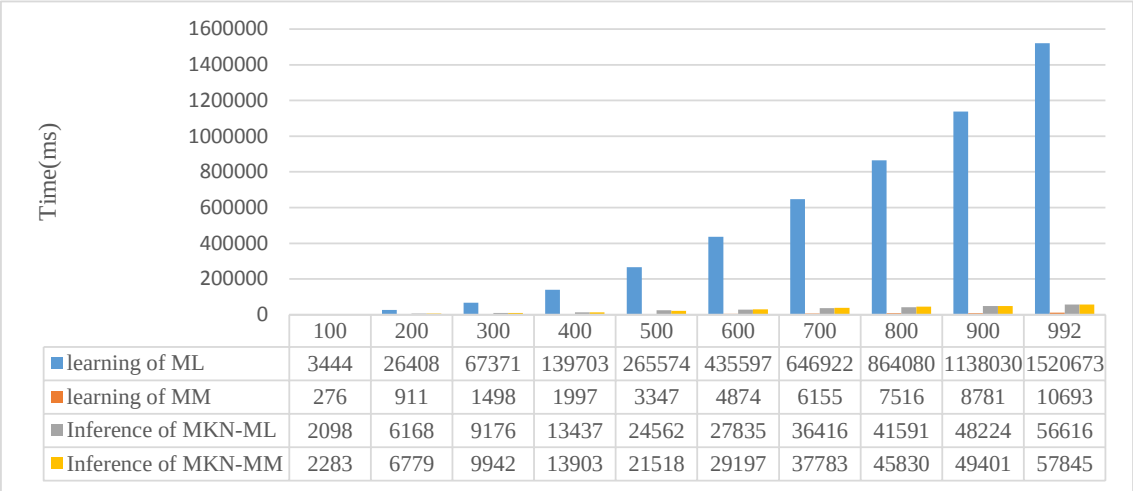
medical knowledge. When more knowledge is learned by the max-margin algorithm and a richer network is built, MKN has more powerful reasoning capability.



	100	200	300	400	500	600	700	800	900	992
Rule Number	125	303	401	493	613	732	832	909	964	1058
Node Number	117	232	301	375	439	502	568	623	647	679
Edge Number	125	303	401	493	613	732	832	909	964	1058

Fig. 6. Structural changes in knowledge network with increasing number of CEMRs.

In addition, an experiment was conducted to analyze the time consumption. In Fig. 7, the first two histograms represent the time consumption of the learning of the maximum-likelihood-based (ML) algorithm and that of the maximum-margin-based (MM) algorithm, respectively. The last two histograms represent the time consumption of the inference of MKN based on the weight knowledge of ML and that based on the weight knowledge of MM, respectively.



	100	200	300	400	500	600	700	800	900	992
learning of ML	3444	26408	67371	139703	265574	435597	646922	864080	1138030	1520673
learning of MM	276	911	1498	1997	3347	4874	6155	7516	8781	10693
Inference of MKN-ML	2098	6168	9176	13437	24562	27835	36416	41591	48224	56616
Inference of MKN-MM	2283	6779	9942	13903	21518	29197	37783	45830	49401	57845

Fig. 7. Time consumption with increasing number of CEMRs.

It is clearly observed that the learning efficiency of the ML algorithm is lower than that of the MM algorithm. As the amount of knowledge increases, ML finds it difficult to complete the learning task within a tolerable time. By contrast, MM could consumes less time and shows better stability. For the inference of MKN, the time consumption can be controlled within a reasonable range, regardless of whether the weight is learned by the ML algorithm or the MM algorithm.

5. Conclusions

In our previous study, we presented an inference algorithm, namely medical knowledge network (MKN), which is a knowledge-based probabilistic model. In the present study, based on the maximum margin criterion of SVMs, we derived a discriminative weight learning algorithm for MKN. After a reasonable margin was defined, we focused on achieving margin optimization by adjusting the knowledge weight. Significantly, we transformed margin optimization into a Lagrange dual problem. By strict derivation of the dual problem, the sequential minimal optimization (SMO) algorithm was adopted for solving two-multiplier sub-problems. Combined with the clique property of a Markov network and the potential

function of MKN, we innovatively developed the inferential process of the finite sum formula. Thus, the knowledge weights could be learned effectively.

In addition, the effectiveness of the max-margin learning algorithm was proved experimentally by comparison with other methods. The empirical results of both machine learning and probabilistic graph models showed that the performance of M^3KN is obviously superior to that of the other methods. This experiment further demonstrated that the max-margin algorithm is more timeliness for learning knowledge weights than the maximum likelihood algorithm. Using the incremental learning method, we finally confirmed that the amount of medical knowledge and the scale of knowledge network have an important influence on disease diagnosis.

In the future, we will study the medical concepts and the relations between them. Furthermore, a massive concept network will be constructed to realize intelligent inference.

Acknowledgement

The Chinese electronic medical records used in this study were provided by the Second Affiliated Hospital of Harbin Medical University. We would like to thank the reviewers for their detailed reviews and insightful comments, which enabled us to improve the quality of this paper.

Appendix A

Fig. A.1 and Fig. A.2 show a progress note and a discharge summary from our records corpus, respectively. Fig. A.3 shows the topology of the knowledge network described in Section 4.4.

```
<?xml version="1.0" encoding="UTF-8"?>
<progress>
    <病例特点>
        1、 既往史、 个人史：吸烟史20余年，平均3支/日.
        2、 现病史：患者3年前起，出现腹痛症状，呈绞痛疼痛，无肩背部放散痛，与进食相关性不确定. 偶有腹胀，
        无恶心，无呕吐，偶觉反酸，无明显发热，间断性入院治疗，症状好转后出院，曾行上消化道内镜检查，
        结果显示慢性胃炎（具体情况不详）. 此次为明确诊治，来我科. 病程中偶有头晕. 以"胃炎"收入院.
        3、 查体：体温36.2℃，脉搏68次/分，呼吸18次/分，血压130/70mmHg. 神志清醒，自主体位、 查体合作.
        全身皮肤正常,巩膜正常，结膜正常. 双肺听诊未闻及干湿啰音，心脏听诊各瓣膜区可闻及病理性杂音区.
        腹部平坦,腹部软，未触及包块，无压痛、 反跳痛，肝脏肋下未及. 脾脏肋下未及移动性浊音阴性.
        肠鸣音4次/分. 四肢活动自如，手末端关节变形. 双下肢及足无凹陷性水肿.
        4、 辅助检查：待回报
    </病例特点>
    <临床初步诊断>
        慢性胃炎
        脑动脉硬化
    </临床初步诊断>
    <诊断依据>
        1、 阵发性上腹部不适3年入院.
        2、 吸烟史20余年，平均3支/日.
        3、 查体：体温36.2℃，脉搏68次/分，呼吸18次/分，血压130/70mmHg. 神志清醒，自主体位、 查体合作.
        全身皮肤正常,巩膜正常，结膜正常. 双肺听诊未闻及干湿啰音，心脏听诊各瓣膜区可闻及病理性杂音区.
        腹部平坦,腹部软，未触及包块，无压痛、 反跳痛，肝脏肋下未及. 脾脏肋下未及移动性浊音阴性.
        肠鸣音4次/分. 四肢活动自如，手末端关节变形. 双下肢及足无凹陷性水肿.
        4、 辅助检查：待回报
    </诊断依据>
    <鉴别诊断>
        1、 胰腺炎:常发生于暴饮暴食或刺激性食物、高蛋白高脂肪食物后,病情进展迅速,腹痛剧烈,血尿淀粉酶显著升高.
        2、 消化性溃疡：临床表现以腹痛为主，疼痛多具有节律性，. 胃镜及病理检查有助于明确诊断.
    </鉴别诊断>
    <诊疗计划>
        1、 胃镜检查病变部位及性质.
        2、 抑制胃酸，保护胃粘膜.
        3、 改善心脑供血.
        4、 支持对症.
    </诊疗计划>
</progress>
```

Characteristics of case

Preliminary clinical diagnosis

Diagnostic basis

Differential diagnosis

Treatment plan

Fig. A.1. A progress note from the records corpus.

```
<?xml version="1.0" encoding="UTF-8"?>
<discharge>
    <住院起止日>
        入院日期:2012-12-03 09:21 出院日期: 2012年12月16日 住院: 13天
    </住院起止日>
    <门诊收治诊断>
        慢性胃炎
    </门诊收治诊断>
    <临床初步诊断>
        慢性胃炎 脑动脉硬化
    </临床初步诊断>
    <临床确定诊断>
        慢性胃炎 脑动脉硬化
    </临床确定诊断>
    <入院时情况>
        女患,主因阵发性上腹部不适3年入院.查体:脉搏66次/分,血压130/70mmHg.无压痛、反跳痛.肝脏肋下未及.
        脾脏肋下未及移动性浊音阴性.肠鸣音次4/分.四肢活动自如,手末端关节变形.双下肢及足无凹陷性水肿.
    </入院时情况>
    <治疗经过>
        1、 胃镜检查病变部位及性质.
        2、 抑制胃酸，保护胃黏膜.
        3、 改善心脑供血.
        4、 支持.
    </治疗经过>
    <出院时情况>
        患者一般状况良好,二便正常.
    </出院时情况>
    <治疗效果>
        好转
    </治疗效果>
    <出院医嘱>
        继续口服药物治疗，定期复查，有不适随诊.
    </出院医嘱>
</discharge>
```

Time periods of hospitalization

Diagnosis of outpatient

Preliminary clinical diagnosis

Definitive clinical diagnosis

Admission condition

Treatment process

Discharge condition

Therapeutic effect

Physician's advice

Fig. A.2. A discharge summary from the records corpus.

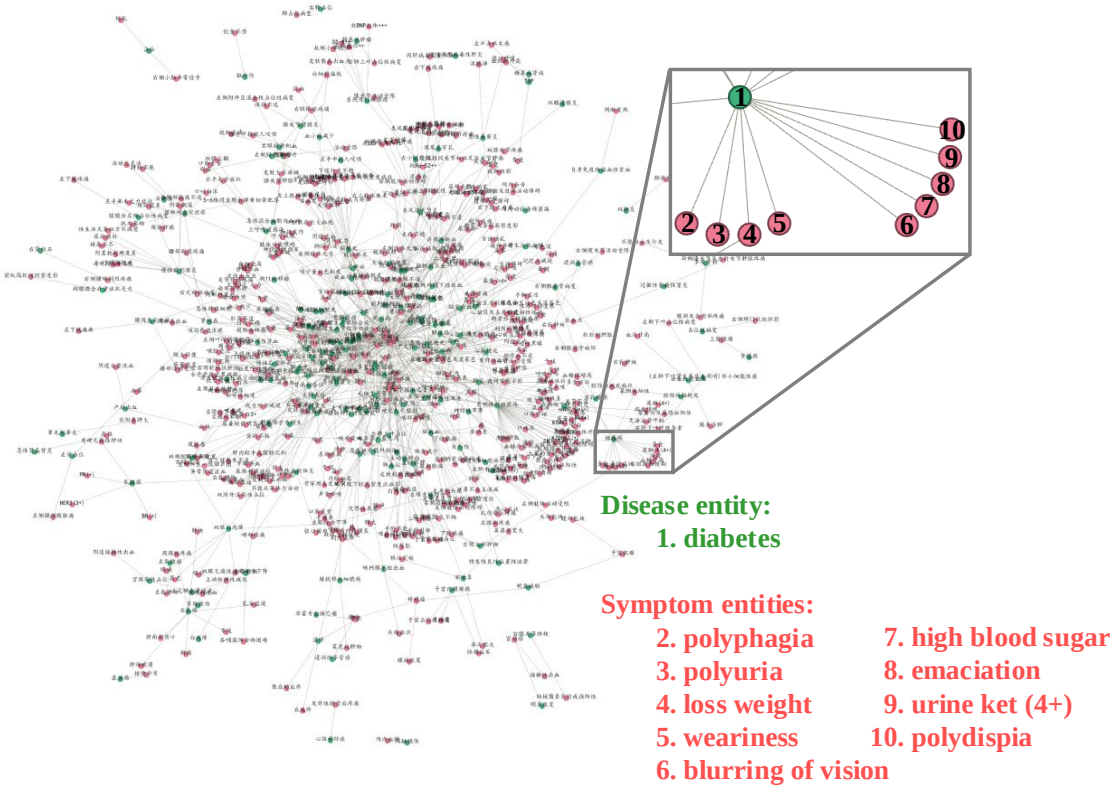


Fig. A.3. Topology of the knowledge network. The nodes in the knowledge network are indicated by different colors according to the type of entity; the red nodes and the green nodes represent "symptom" entities and "disease" entities, respectively.

For the reader's benefit, we provide the entire derivation of the SMO learning of M^{3}KN: (1) the max-margin problem is transformed into a quadratic programming problem and the basic Lagrange function is defined; (2) a complex expansion formula of $\max_{\alpha}\min_{\omega} L$ is derived; and (3) the expression of Lagrange multiplier α_1 is substituted into $\max_{\alpha}\min_{\omega} L$.

(1) By solving the objective function $\gamma(x, y, \omega)$, we define the complete form of the Lagrange function L:

$$L = \frac{1}{2}\omega^T \sum_{i=1}^{n} \alpha_i (x_i y_i - x_i \hat{y}_i) - \sum_{i=1}^{n} \alpha_i \omega^T (x_i y_i - x_i \hat{y}_i) + \sum_{i=1}^{n} \alpha_i \Delta t - \sum_{i=1}^{n} \alpha_i \varepsilon_i + C \sum_{i=1}^{n} \varepsilon_i . \quad (17)$$

By setting $\partial L/\partial \omega$ to zero, we can obtain $\omega = \sum_{i=1}^{n} \alpha_i (x_i y_i - x_i \hat{y}_i)$. Then, we define $K_i = x_i y_i - x_i \hat{y}_i$. The entire mathematical deduction of Eq. (7) is given by:

$$\begin{aligned}
\min_{\omega} L = -\max_{\omega} L &= -\frac{1}{2}\omega^T \sum_{i=1}^{n} \alpha_i (x_i y_i - x_i \hat{y}_i) + \sum_{i=1}^{n} \alpha_i \omega^T (x_i y_i - x_i \hat{y}_i) - \sum_{i=1}^{n} \alpha_i \Delta t + \sum_{i=1}^{n} \alpha_i \varepsilon_i - C \sum_{i=1}^{n} \varepsilon_i \\
&= \frac{1}{2}\omega^T \sum_{i=1}^{n} \alpha_i (x_i y_i - x_i \hat{y}_i) - \sum_{i=1}^{n} \alpha_i \Delta t + \sum_{i=1}^{n} \alpha_i \varepsilon_i - C \sum_{i=1}^{n} \varepsilon_i \\
&= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j (x_i y_i - x_i \hat{y}_i)(x_j y_j - x_j \hat{y}_j) - \sum_{i=1}^{n} \alpha_i \Delta t \\
&= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K_i K_j - \sum_{i=1}^{n} \alpha_i \Delta t \\
&s.t., \quad \sum_{i=1}^{n} \alpha_i = C \\
&\qquad 0 \le \alpha_i \le C
\end{aligned} \tag{18}$$

(2) To simplify the objective function $\max_{\alpha} \min_{\omega} L$, we expand two corresponding additive terms into the polynomial form of the variable and obtain the following solutions:

$$\begin{aligned}
\max_{\alpha} \min_{\omega} L &= \frac{1}{2} \sum_{i=1}^{n} [\alpha_i \alpha_1 K_i K_1 + \alpha_i \alpha_2 K_i K_2 + \sum_{j=3}^{n} \alpha_1 \alpha_j K_i K_j] - \sum_{i=1}^{n} \alpha_i \Delta t \\
&= \frac{1}{2} [\alpha_1^2 K_1^2 + \alpha_1 \alpha_2 K_1 K_2 + \sum_{j=3}^{n} \alpha_1 \alpha_j K_1 K_j + \alpha_1 \alpha_2 K_1 K_2 + \alpha_2^2 K_2^2 + \sum_{j=3}^{n} \alpha_2 \alpha_j K_2 K_j \\
&\quad + \sum_{i=3}^{n} \alpha_1 \alpha_j K_1 K_j + \sum_{i=3}^{n} \alpha_2 \alpha_j K_2 K_j + \sum_{i=3}^{n} \sum_{j=3}^{n} \alpha_i \alpha_j K_i K_j] - \sum_{i=3}^{n} \alpha_i \Delta t - \alpha_1 \Delta t - \alpha_2 \Delta t . \\
&= \frac{1}{2} [\alpha_1^2 K_1^2 + 2\alpha_1 \alpha_2 K_1 K_2 + \alpha_2^2 K_2^2 + 2\sum_{i=3}^{n} \alpha_1 \alpha_i K_1 K_i + 2\sum_{i=3}^{n} \alpha_2 \alpha_i K_2 K_i \\
&\quad + \sum_{i=3}^{n} \sum_{j=3}^{n} \alpha_i \alpha_j K_i K_j] - \sum_{i=3}^{n} \alpha_i \Delta t - \alpha_1 \Delta t - \alpha_2 \Delta t
\end{aligned} \tag{19}$$

(3) According to the constraint conditions of the Lagrange multiplier from Eq. (9), Lagrange multiplier α_2 is held for Eq. (19), and it can be rewritten as:

$$\begin{aligned}\max_{\alpha}\min_{\omega} L &= \frac{1}{2}[(T-\alpha_2)^2K_1^2 + 2(T-\alpha_2)\alpha_2K_1K_2 + \alpha_2^2K_2^2 + 2\sum_{i=3}^{n}(T-\alpha_2)\alpha_iK_1K_i \\ &\quad + 2\sum_{i=3}^{n}\alpha_2\alpha_iK_2K_i + \sum_{i=3}^{n}\sum_{j=3}^{n}\alpha_i\alpha_jK_iK_j] - \sum_{i=3}^{n}\alpha_i\Delta t - (T-\alpha_2)\Delta t - \alpha_2\Delta t \\ &= \frac{1}{2}[(T-\alpha_2)^2K_1^2 + 2T\alpha_2K_1K_2 - 2\alpha_2^2K_1K_2 + \alpha_2^2K_2^2 + 2\sum_{i=3}^{n}T\alpha_iK_1K_i \\ &\quad - 2\sum_{i=3}^{n}\alpha_2\alpha_iK_1K_i + 2\sum_{i=3}^{n}\alpha_2\alpha_iK_2K_i + \sum_{i=3}^{n}\sum_{j=3}^{n}\alpha_i\alpha_jK_iK_j] - \sum_{i=3}^{n}\alpha_i\Delta t - T\Delta t\end{aligned} \quad . \qquad (20)$$

References

[1] S. Belciug, F. Gorunescu. Error-correction learning for artificial neural networks using the Bayesian paradigm. Application to automated medical diagnosis. J. Biomed. Inform., 2014, 52: 329-337.

[2] F. Ahmad, N.A.M. Isa, Z. Hussain, et al. Intelligent medical disease diagnosis using improved hybrid genetic algorithm-multilayer perceptron network. J. Med. Syst., 2013, 37(2): 9934.

[3] R. Alizadehsani, J. Habibi, M.J. Hosseini, et al. A data mining approach for diagnosis of coronary artery disease. Comput. Methods Programs Biomed., 2013, 111(1): 52-61.

[4] M. Hariharan, K. Polat, R. Sindhu. A new hybrid intelligent system for accurate detection of Parkinson's disease. Comput. Methods Programs Biomed., 2014, 113(3): 904-13.

[5] C. Zhao, J. Jiang, Z. Xu, et al. A study of EMR-based medical knowledge network and its applications. Comput. Methods Programs Biomed., 2017, 143: 13-23.

[6] D. Vassis, B.A. Kampouraki, P. Belsis, et al. Using neural networks and SVMs for automatic medical diagnosis: a comprehensive review. AIP Conference Proceedings. AIP, 2015, 1644(1): 32-36.

[7] Jiang J, Zheng J, Zhao C, et al. Clinical-decision support based on medical literature: A complex network approach[J]. Physica A Statistical Mechanics & Its Applications, 2016, 459:42-54.

[8] T. Palmerini, U. Benedetto, L. Bacchi-Reggiani, et al. Mortality in patients treated with extended duration dual antiplatelet therapy after drug-eluting stent implantation: a

pairwise and Bayesian network meta-analysis of randomised trials. Lancet, 2015, 385(9985): 2371-2382.

[9] L. Snidaro, I. Visentini, K. Bryan. Fusing uncertain knowledge and evidence for maritime situational awareness via Markov Logic Networks. Inf. Fusion, 2015, 21: 159-172.

[10] S. Marini, E. Trifoglio, N. Barbarini, et al. A Dynamic Bayesian Network model for long-term simulation of clinical complications in type 1 diabetes. J. Biomed. Inform., 2015, 57: 369-376.

[11] P. Fuster-Parra, P. Tauler, M. Bennasar-Veny, et al. Bayesian network modeling: A case study of an epidemiologic system analysis of cardiovascular risk. Comput. Methods Programs Biomed., 2016, 126: 128-142.

[12] M. Richardson, P. Domingos. Markov logic networks. Mach. Learn., 2006, 62(1-2): 107-136.

[13] S. Natarajan, V. Bangera, T. Khot, et al. Markov logic networks for adverse drug event extraction from text. Knowl. Inf. Syst., 2016: 1-23.

[14] Y.C. Song, H. Kautz, J. Allen, et al. A Markov logic framework for recognizing complex events from multimodal data. Proceedings of the 15th ACM International Conference on Multimodal Interaction. ACM, 2013: 141-148.

[15] C. Li, P. Zhao, J. Wu, et al. Anatomy ontology matching using Markov logic networks. Scientifica, 2016, 2016.

[16] K. Kalidoss, K. Vajravelu. PROCEOL: Probabilistic relational of concept extraction in ontology learning. IRECOS, 2014, 9(4): 716-726.

[17] H. Poon, P.M. Domingos. Machine Reading: A" Killer App" for Statistical Relational AI. Statistical Relational Artificial Intelligence, 2010, 10: 06.

[18] Z. Liu, G. von Wichert. A generalizable knowledge framework for semantic indoor mapping based on Markov logic networks and data driven MCMC. Future Gener. Comput. Syst., 2014, 36: 42-56.

[19] T. Khot, S. Natarajan, K. Kersting, et al. Gradient-based boosting for statistical relational learning: the Markov logic network and missing data cases. Mach. Learn., 2015, 100(1): 75-100.

[20] J. van Haaren, G. van den Broeck, W. Meert, et al. Lifted generative learning of Markov logic networks. Mach. Learn., 2016, 103(1): 27-55.

[21] A. Nath, P.M. Domingos. Learning Relational Sum-Product Networks. AAAI. 2015: 2878-2886.

[22] B. Geng, X. Zhou, J. Zhu, et al. Comparison of reversible-jump Markov-chain-Monte-Carlo learning approach with other methods for missing enzyme identification. J. Biomed. Inform., 2008, 41(2): 272-281.

[23] B. Roller. Max-margin Markov networks. Adv. Neural Inf. Process. Syst., 2004, 16: 25.

[24] T.N. Huynh, R.J. Mooney. Max-margin weight learning for Markov logic networks. Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg, 2009: 564-579.

[25] T.N. Huynh, R.J. Mooney. Online max-margin weight learning for Markov logic networks. Proceedings of the 2011 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2011: 642-651.

[26] PRIMARY PSYCHIATRY, "Electronic Medical Records." [Online] Available: http://primarypsychiatry.com/electronic-medical-records/.

[27] A.K. Sari, W. Rahayu, M. Bhatt, Archetype sub-ontology: Improving constraint-based clinical knowledge model in electronic health records, Knowl.-Based Syst., 2012, 26(1): 75-85.

[28] S.L. Ting, S.K. Kwok, A.H.C. Tsang, et al., A hybrid knowledge-based approach to supporting the medical prescription for general practitioners: Real case in a Hong Kong medical center, Knowl.-Based Syst., 2011, 24(3): 444-456.

[29] U. Iqbal, C.K. Hsu, P.A. Nguyen, et al. Cancer-disease associations: A visualization and animation through medical big data. Comput. Methods Programs Biomed., 2016, 127: 44-51.

[30] S. Keretna, C.P. Lim, D. Creighton, et al. Enhancing medical named entity recognition with an extended segment representation technique. Comput. Methods Programs Biomed., 2015, 119(2): 88-100.

[31] H. López-Fernández, M. Reboiro-Jato, D. Glez-Peña, et al. BioAnnote: a software platform for annotating biomedical documents with application in medical learning

environments. Comput. Methods Programs Biomed., 2013, 111(1): 139-147.

[32] WILAB-HIT, "Resources." [Online] Available: https://github.com/WILAB-HIT/Resources/.

[33] P. Singla, P. Domingos. Discriminative training of Markov logic networks. AAAI. 2005, 5: 868-873.

[34] J.C. Jiang, C. Zhao, Y. Guan, Q.B. Yu. Learning and inference in knowledge-based probabilistic model for medical diagnosis. arXiv preprint arXiv:1703.09368, 2017.

[35] I. Sutskever, G.E. Hinton, G.W. Taylor. The recurrent temporal restricted Boltzmann machine. Adv. Neural Inf. Process. Syst., 2009: 1601-1608.

[36] I2B2, "Informatics for Integrating Biology & the Bedside." [Online] Available: https://www.i2b2.org/.

[37] G.E. Hinton, T.J. Sejnowski, Optimal perceptual inference, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (1983) 448-453.

[38] E. Ising, Beitrag zur theorie des ferromagnetismus, Zeitschrift für Physik A Hadrons and Nuclei. 31(1) (1925) 253-258.

[39] F. Pernkopf, M. Wohlmayr, S. Tschiatschek. Maximum margin Bayesian network classifiers. IEEE T. Pattern Anal., 2012, 34(3): 521-532.

[40] S. Lacoste-Julien. Combining SVM with graphical models for supervised classification: an introduction to Max-Margin Markov Network. CS281A Project Report, UC Berkeley, 2003.

[41] C.C. Chang, C.J. Lin. LIBSVM: a library for support vector machines. ACM TIST, 2011, 2(3): 27.

[42] Ö. Uzuner, Y. Luo, P. Szolovits, Evaluating the state-of-the-art in automatic de-identification, J. Am. Med. Inform. Assn., 14(5) (2007) 550-563.

[43] B. He, Y. Guan, J. Cheng, et al. CRFs based de-identification of medical records. J. Biomed. Inform., 2015, 58: S39-S46.

[44] S. Zhang, N. Elhadad. Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. J. Biomed. Inform., 2013, 46(6): 1088-1098.

[45] F. Niu, C. Ré, A.H. Doan, et al. Tuffy: Scaling up statistical inference in Markov logic networks using an RDBMS. Proceedings of the VLDB Endowment, 2011, 4(6): 373-384.

Publication 9

HIT-WI at TREC 2015 clinical decision support track

Jingchi Jiang, Yi Guan, Jia Su, Chao Zhao, and Jinfeng Yang

HIT-WI at TREC 2015 Clinical Decision Support Track

Jingchi Jiang[1], Yi Guan[1*], Jia Su[1], Chao Zhao[1], Jinfeng Yang[2]

[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
[2]School of Computer Science and Technology, Harbin University of Science and Technology, Harbin, 150080, China

jiangjingchi0118@163.com, guanyi@hit.edu.cn, sjd163mail@163.com, hitsa.zc@gmail.com, fondofbeyond@163.com

Abstract. The TREC 2015 Clinical Decision Support track is composed of two subtasks, task A and task B. Similar to 2014 [1], the participants need to answer 30 clinical questions from patient cases for each task. According to the three types of clinical question: diagnosis, test and treatment, these tasks are to retrieve relevant literatures for helping clinicians to make clinical decision.

This paper describes how the clinical decision support system is developed for completing the task A and B by the HIT-WI group. For the automatic runs, some classical retrieval strategies are adopted, including query extraction, query expansion and the process of retrieval. Moreover, we propose two novel re-ranking methods: the one uses SVM model with 10-dimensional feature to re-rank the retrieved list, and the other is based on word co-occurrence network.

The 178 runs are submitted from 36 different groups. Our evaluation results show that 1) The Indri performs better than Lucene's for artificially-constructed queries. 2) Compare to the basic retrieval method, two re-ranking methods show the effectiveness in some topics. 3) Our results are higher than the median scores in most topics of task B. Furthermore, the system achieves the best scores for topics: #11 and #12.

1 Introduction

As a hot spot of academic frontier, Clinical decision support (CDS) provides clinicians and health professionals with knowledge and personalized information at appropriate times, to enhance the health level of patients. In making clinical decisions, clinicians often review the medical literature to further ensure the reliability for diagnosis and treatment. Medical literature can answer the three most common generic clinical questions faced by clinicians everyday [2]:"what is the patient's diagnosis?", "what tests should the patient receive?", "how should the patient be treated?". However, the problem of retrieving the relevant literatures can be time-consuming and difficult under the circumstance of massive literatures.

Similar to the goal of 2014, the TREC 2015 Clinical Decision Support (CDS) track is designed to retrieve relevant medical articles for answering generic clinical questions, according to actual patient records [3]. A patient record typically describes a

* Corresponding author. Tel:+86 18686748550. E-mail: guanyi@hit.edu.cn

challenging medical case, and mainly contains two sections: description which describes patients' condition in detail and summary, which synthesizes meaningful information from description based on the experience of doctors. The corpus for the retrieval task is the Open Access Subset of PubMed Central (PMC) on January 21, 2014, which contains a total of 733,138 articles [4].

In this paper, traditional retrieval techniques are adopted [5], including medical terms extraction, query expansion and literature retrieval. Then, we propose two re-ranking methods to enhance the relevance of retrieved results.

The rest of this paper is arranged as follows. In Sec. 2, we discuss the materials and methods in detail, and also focus on the construction of re-ranking models. Moreover, we conduct the experiments to testify the effectiveness of clinical decision system in Sec. 3. In Sec. 4, we conclude this paper and discuss the directions for further work.

2 Methods

2.1 Query construction

The query construction consists of query extraction, query expansion and query set generation. In the process of query's auto-construction, Metamap (a tool to map bio-medical text to the UMLS Metathesaurus) is used for extracting the medical concepts from the summary section of patient records. In addition, some rules are established, according to whether the concept's semantic type belongs to what we summarize, such as Neoplastic Process, Sign or Symptom et al. Then we regard these filtered medical concepts as the basic query set.

However, the basic queries which are only extracted from the given patient record, cannot completely retrieve relevant literatures for answering the clinical questions. Therefore, we adopt the UMLS Metathesaurus to expand the concepts. In the process of expanding, we avoid the same words presented in query as much as possible and add the type words (diagnosis, test and treatment) for improving the accuracy.

After a series of steps, the query sets are generated automatically in a different formats, to fit the different search engines.

2.2 The process of retrieval

The PubMed Central articles are published in the form of XML, one file per article. Therefore, an XML parser is employed to extract PMC ID, keyword, title, abstract, body and reference from each article.

In order to compare the retrieval performance of search engine, we adopt two kinds of toolkits: Intri and Apache Lucene, respectively. The former provides state-of-the-art text search and a rich structured query language. The latter is based on language model approach with Jelinek-Mercer smoothing for retrieving articles.

We start to retrieve the relevant literatures, including query extraction and expansion, building index. Each participant can only submit 1000 literatures at most for each topic. Therefore, we select the top 1000 literatures as the final result, which is ranked as the given score by the search engine.

2.3 Re-ranking model

2.3.1 Re-ranking based on machine learning

According to the relevant results of TREC 2014, it becomes possible for us to use machine learning method to re-rank the retrieved list. To judge whether a literature is relevant to the clinical decision, we think empirically that the appearing position of a query is a significant feature. Because, it is reasonable that the query appears in title of literature is more important than the same query appears in body. In addition, the position of type words, such as diagnosis, test or treatment, is also a strong feature to judge the relevance.

Due to a literature contains five fields, including title, abstract, keywords, body and reference, we extract the query feature and the type word feature from each fields, respectively. Therefore, a total of 10 features will be extracted for a certain literature.

We construct queries from TREC 2014 topics and obtain the retrieved results. Every retrieved literature is labeled as 0, 1 and 2, which represent the non-relevance, possible relevance and completely relevance. However, considering the quantity of relevant literatures is far less than the quantity of irrelevant ones, we regard the label 1 and 2 as relevant. Using the SVM classifier with a linear kernel to classify relevant literatures from irrelevant ones, the SVM model is trained. Then this model is applied to retrieved results of TREC 2015. Every result would be labeled either 1 if relevant, or 0 if not relevant. Adding this score with a 0.25 gain to the original indri score, we obtain the new score, which is used for our re-ranking.

2.3.2 Re-ranking based on co-occurrence network

In order to improve the performance of relevance ranking, we propose a novel method to re-rank the retrieved results. The idea of this method is based on co-occurrence words. We build a co-occurrence network to mine the potential literatures. For improving the recall rate, the re-ranking formula is constructed based on some network features.

2.3.2.1 The construction of co-occurrence network

In the process of analysis for the TREC 2014, we find that these relevant literatures have a lot of co-occurrence words. We assume that these co-occurrence words can reveal the relevance of literature. In order to validate this assumption, an intuitive co-occurrence network based on 1000 retrieved literatures is needed. Firstly, we empirically extract the co-occurrence words from literature by the top level of MeSH hierarchy [6]:

- Diagnosis: B03, B04, C
- Test: E01
- Treatment: D02, D04, D06, D26, D27, E02, E04

When a common medical word from MeSH appears on two literatures, an edge will be created to connect them. Moreover, the edge weight gradually grow, along

with the number of common words increasing. After 1000 retrieved literatures are iterated, a co-occurrence network is built, which is composed of the literature as the node and the co-occurrence medical word as the edge. The topology of the co-occurrence network is shown in Fig. 1.

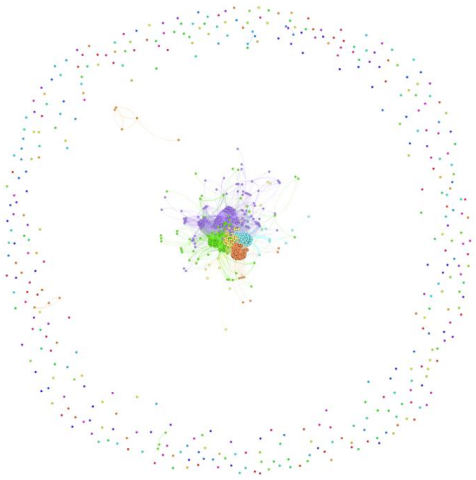


Fig. 1 The topology of co-occurrence network

As shown in Fig.1, the network consists of several communities in different color. The features of the co-occurrence network include the followings: 1. The literature nodes within the same community are strongly attached to each other. 2. Instead, the nodes from different communities represent a "weaker" relation. Through observing and analyzing the judgment file of TREC 2014, we find that the most of relevant literature locate the inside of community, while the discrete nodes always play the non-relevance or low relevance roles for each topic. Furthermore, we can summarize that every community have dissimilar emphases for the given patient record. The subject of some communities are appropriate to answer the clinical question, while the literatures from the other communities are irrelevant. Therefore, how to choose the appropriate communities is an important work.

2.3.2.2 Mining potential literatures

Because the automatic extraction and expansion have its limitations and uncertainties, lead to the incomplete and non-credibility of query set. Therefore, some relevant literatures might be missed except 1000 retrieved literatures. To solve this problem, we propose a method based on the co-occurrence network, to mining potential relevant literatures from the rest of the corpus. This method uses the indicator of clustering coefficient to determine whether a literature is associated with the topic.

Node coefficient is defined as the proportion of connections among its neighbors which are actually realized compared with the number of all possible connections. The parameter *k* is defined as the number of the common terms from MeSH between

literature i and community ζ. $T(i)$ represents the number of all possible connections among the k vertices.

$$T(i) = k(k-1)/2 \tag{1}$$

$E(i)$ represents the actual number of edges among the k vertices. $c(i)$ is the clustering coefficient of node i and can be computed as follows.

$$c(i) = E(i)/T(i) \tag{2}$$

Community coefficient is defined as the mean of the entire node coefficient within the community. $c(\zeta)$ is defined as the clustering coefficient of community ζ :

$$c(\zeta) = \frac{\sum_{i-1}^{n} c(i)}{n} \tag{3}$$

If the node coefficient is greater than the community coefficient of a specific community, we can conclude that this node has similarity to this community, and put the node as a potential literature to the existing community. The bigger node coefficient means that the higher connectivity with the community. Along with the continuous increase of the potential literatures, some new MeSH terms will be found from the co-occurrence network, which can describe the topic better. After all of the literatures are traversed, a richer co-occurrence network is built.

2.3.2.3 Re-ranking calculation

Based on the richer co-occurrence network, we need to re-rank all the nodes. To calculate the score of re-ranking, some measures should be introduced, including the measure of node importance and the medical terms density of community where the node locate.

Because the potential literatures are different than the retrieved literatures which have a relevance score by search engine. Therefore, a computational method for calculating the relevance score of potential literature is also proposed, which is defined as follows:

$$Score(i) = \begin{cases} Rscore(i) & i \in RetrievedSet \\ NC(i) * \sum_j Rscore(j)/n & i \in PotentialSet \end{cases} \tag{4}$$

$Rscore(i)$ represents the relevance score of retrieved literature by search engine. $NC(i)$ is the node coefficient of literature i. j is defined as a literature within retrieved set, while is connected to the literature i. n is the number of literature j.

The terms density of community is defined as the ratio of the number of terms to the number of relationships. In addition, we adopt the value of pagerank to regard as

the measure of node importance. After the preparation of the theory, we propose the formula of re-ranking model:

$$ReRankScore(i) = \alpha \cdot CD_i \cdot PR(i) + \beta \cdot Score(i) \quad (5)$$

CD_i represents the density of community where is the literature i location. $PR(i)$ is the importance of literature i. α and β are both the weight parameters for regulating the factor proportion between the co-occurrence network and the search engine.

3 Experiments

3.1 Clinical decision support system design

Our clinical decision support system consists of four main modules.

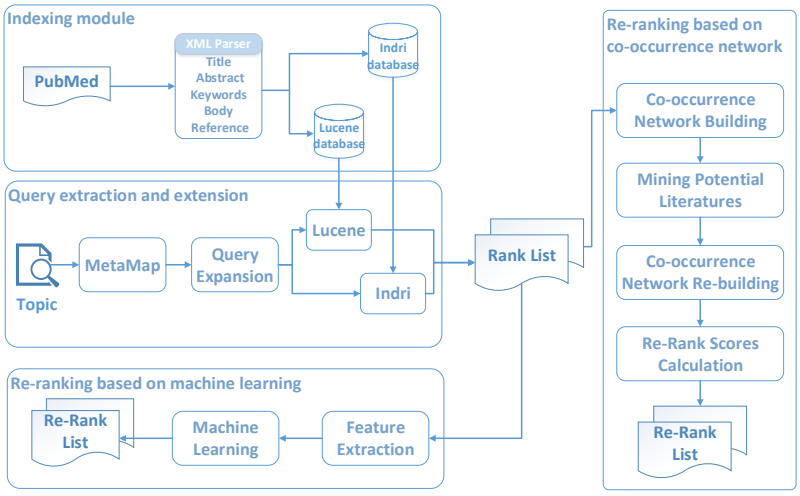


Fig. 1. The flow diagram of the clinical decision support system.

3.2 Comparing Indri with Lucene

In the TREC 2015 Clinical Decision Support track, the task consists of two parts: the task A and task B which adds the "diagnosis" section from the last twenty topics. For the task A, we submit the retrieved results including artificial Indri result, automatic Indri result and automatic Indri result with the re-ranking model based on machine learning. Similar with task A, the artificial list of Lucene, automatic list of Indri and the automatic result with the co-occurrence network are submitted for the task B.

Because the topics of diagnosis type have exactly the same contents and structures in task A and B. So we can compare the result of artificial Indri and artificial Lucene based on the same query set. Figure 2 shows the difference between Indri and Lucene

using four different measurement indicators. We can see that the former outperforms the latter in most of topic. It follows that the search engine of Indri is more effective than Lucene for the retrieval task.

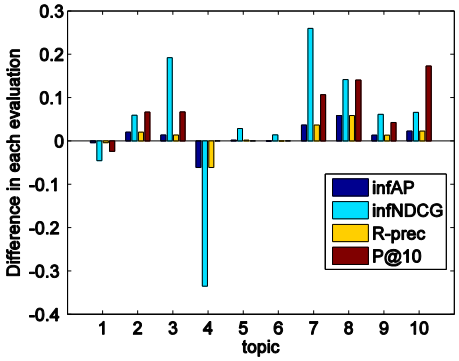


Fig. 2. Comparing Indri with Lucene.

3.3 Comparing submitted runs to each other

In order to testify the effectiveness of our methods, we compare the infAP and infNDCG of each method. For the task A as shown as Figure. 3, the artificial results are much higher than other automatic results. It is also find that the re-ranking model based on machine learning has less effective than the expectations.

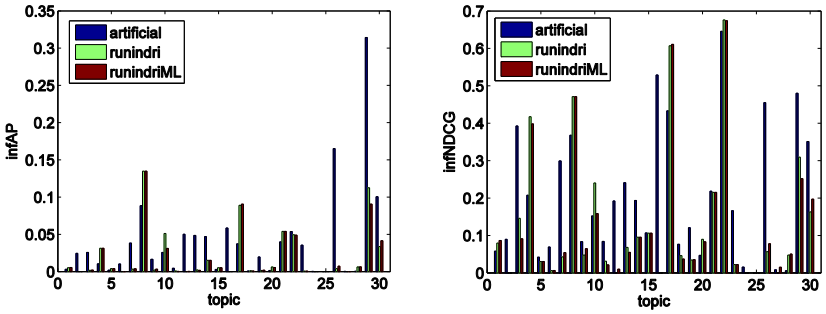


Fig. 3. Retrieval results for task A in two different indicators.

From the statistical results of Figure. 4, the re-ranking model based on co-occurrence network does not perform well enough. The performance of most topics is not improved except a small rise in the topic 9 and 27. For the possible reasons of unsatisfactory result, we analyze that it could be caused by the weight parameters of α and β, which are not adjusted to the optimal values.

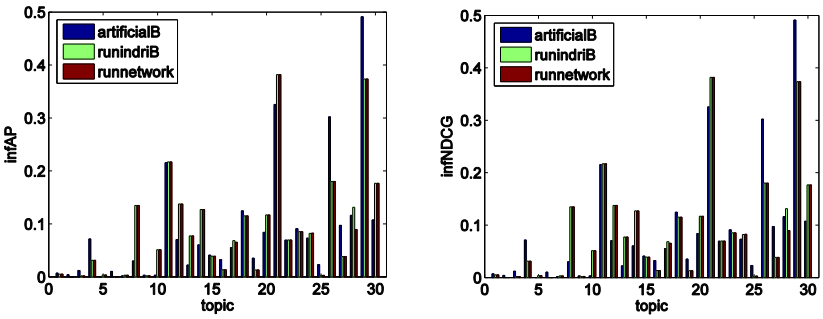


Fig. 4. Retrieval results for task B in two different indicators.

3.4 Comparing submitted runs to the median

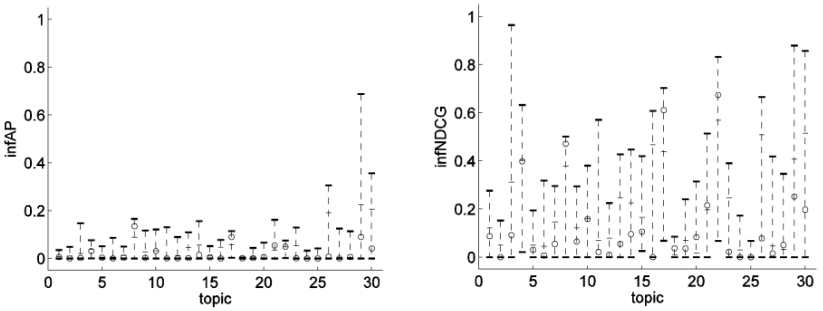


Fig. 5. Comparing the automatic runs based on SVM model with other participants.

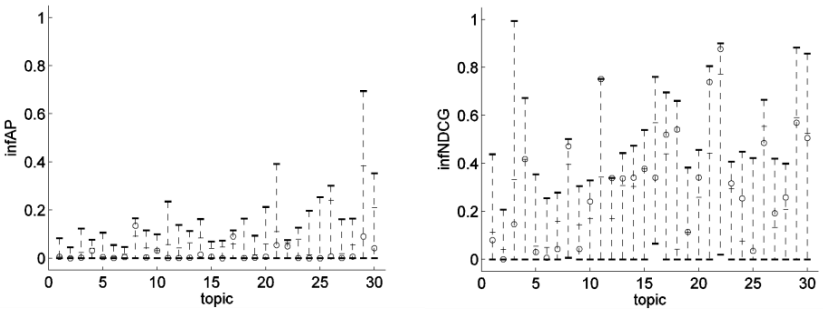


Fig. 6. Comparing the automatic runs based on co-occurrence network with other participants.

Finally, a set of experimental results is given. The automatic runs with re-ranking model based on SVM are below the median scores for the most topics, as shown in Figure. 5. From the Figure. 6, we can see that our re-ranking model based on co-

occurrence network achieve the best score in two topics: #11 and #12. The model also performs much better than the median scores for the other topics. These results further testify the effectiveness of our clinical decision support system.

4 Conclusion

This paper described the clinical decision support task in the TREC 2015. To complete the task, a clinical decision support system based on literatures is designed and developed by the HIT-WI group. On the basis of traditional retrieval techniques, we propose two novel re-ranking methods to improve the retrieval results. The two methods use the models of the machine learning and the network. Moreover, the analysis of the experimental result demonstrates the effectiveness of our system. Our future work will focus on optimizing the re-ranking model and cutting down time consumption in the process of retrieval.

Acknowledgments. The Open Access Subset of PubMed Central used in this paper were provided by TREC 2015 Clinical Decision Support (CDS) track, and thanks to the organizing committee of TREC and the evaluators of the results.

References

1. Simpson M S, Voorhees E, Hersh W. Overview of the TREC 2014 Clinical Decision Support Track[C]//Proc. 23rd Text Retrieval Conference (TREC 2014). National Institute of Standards and Technology (NIST). 2014.
2. Hasan S A, Zhu X, Dong Y, et al. A Hybrid Approach to Clinical Question Answering[J].
3. Gobeillab J, Gaudinata A, Paschec E, et al. Full-texts representation with Medical Subject Headings, and co-citations network rerank-ing strategies for TREC 2014 Clinical Decision Support Track[J].
4. Wei Y, Hsu C C, Thomas A, et al. Atigeo at TREC 2014 Clinical Decision Support Task[J].
5. Garcia-Gathrighta J I, Menga F, Hsua W. UCLA at TREC 2014 Clinical Decision Support Track: Exploring Language Models, Query Expansion, and Boosting[J].
6. Mourao A, Martins F, Magalhaes J. NovaSearch at TREC 2014 Clinical Decision Support Track[J].

Publication 10

WI-ENRE in CLEF eHealth Evaluation Lab 2015: clinical named entity recognition based on CRF

Jingchi Jiang, Yi Guan, and Chao Zhao

WI-ENRE in CLEF eHealth Evaluation Lab 2015: Clinical Named Entity Recognition Based on CRF

Jingchi Jiang[1], Yi Guan[1], Chao Zhao[1]

[1]School of Computer Science and Technology,
Harbin Institute of Technology, Harbin, China
jiangjingchi0118@163.com, guanyi@hit.edu.cn, hitsa.zc@gmail.com

Abstract. Named entity recognition of biomedical text is the shared task 1b of the 2015 CLEF eHealth evaluation lab, which focuses on making biomedical text easier to understand for patients and clinical workers. In this paper, we propose a novel method to recognize clinical entities based on conditional random fields (CRF). The biomedical texts are split into sections and paragraphs. Then the NLP tools are used for POS tagging and parsing, and four groups of features are extracted to train the entity recognition model. In the subsequent phase for entity normalization, the MetaMap of Unified Medical Language System (UMLS) tool is used to search for concept unique identifiers (CUIs) category. In addition, CRF++ package is adopted to recognize clinical entities in another phase for entity recognition. The experiments show that our system named as WI-ENRE, is effective in the named entity recognition of biomedical texts. The $F_{measure}$ of EMEA and MEDLINE reach to 0.56 and 0.45 respectively in exact match.

Keywords: Named Entity Recognition, Conditional Random Fields, UMLS

1 Introduction

With the application of EMRs, hospitals and medical institutions generate masses of biomedical text. Based on biomedical text, the medical big data analytics and the building of heath knowledge network are the critical problem. As a precondition to solve the problem, named entity recognition can provide a solution to extract information and knowledge from biomedical text. Hence, the named entity recognition is becoming a research hotspot.

Biomedical text contains a wealth of information on patients covering their hospital stays, including health conditions, diagnoses, performed tests and treatments. Named entity recognition form biomedical text has a good research foundation[1,2]. In previous years, several NLP shared tasks have addressed information extraction tasks such as 2010 i2B2/VA Challenge[3] as well as identifying protected health information (PHI) at 2014 i2b2/UTHealth challenge. The 2013 ShARe/CLEF eHealth T2 task[4] was required to detect disorders spans and their concept unique identifiers (CUIs). On that basis, the 2014 ShARe/CLEF eHealth T2 shared task[5] focused on extracting information from biomedical text. In 2015, the CLEFeHealth addresses clinical

named entity recognition on task 1b. The aim is to automatically identify clinically relevant entities in medical text with French rather than English.

Methods for entity recognition can be roughly divided into three categories: rule-based, machine learning methods and a combination of both. The method of rule-based mainly relies on proper nouns dictionaries and rules which wrote by language experts or domain experts to identify the clinical entities. Compared to rule-based methods, many more researchers choose machine learning methods on entity recognition.

In this paper, we propose a novel method for task 1b of CLEFeHealth 2015. In order to testify this method, we design a named entity recognition system, WI-ENRE, which adopts machine learning method based on conditional random fields for the nine categories and lexicon-based approach for geographic areas.

The rest of this paper is arranged as follows. In Sec. 2, we discuss the materials and methods in detail, and also focus on feature optimizing selection. Moreover, we conduct the experiments to testify the effectiveness of WI-ENRE in Sec. 3. In Sec. 4, we conclude this paper and discuss the directions for further work.

2 Methods

In this study, the dataset which is called QUAERO French Medical Corpus[6] is provided by 2015 CLEFeHealth shared tasks. The training set consists of 11 text files with corresponding annotation files from EMEA and 833 text files with annotation files from MEDLINE. 80% of the text files from MEDLINE and EMEA folders are selected as the training data of model, respectively, while the remaining files are used for testing.

In the process of entity recognition and entity normalization, some related resources are used, which contain Stanford Parser based on French and UMLS tool. Then, the feature selection will be described as the significant part in this paper. Finally, the principle of conditional random field algorithm will be detailed in Sec. 2.4.

2.1 Data

The corpus is provided by the 2015 CLEFeHealth evaluation lab. The task 1b consists of clinical named entity recognition and entity normalization from the file of MEDLINE titles and EMEA documents.

In order to testify the method of entity recognition, the training set provided by CLEFeHealth is divided into two parts: the dataset for training which contains 676 documents and a total of 22,160 words, and the testing set contains 168 documents and a total of 3,336 words. Moreover, the number of entity and deduplicated entity are counted, respectively (as shown Tab. 1). In Tab. 2, we also give a few statistics for each category in the training corpus.

Table 1. Description of the corpus.

	Training	Test
MEDLINE Documents	667	166
EMEA Documents	9	2
MEDLINE Words	8,406	2,149
EMEA Words	13,754	1,187
MEDLINE Entities	2,383	612
EMEA Entities	2,357	338
MEDLINE Entities(Deduplication)	1,879	541
EMEA Entities(Deduplication)	848	166

Table 2. Statistics of each category from the training corpus.

Category	MEDLINE	EMEA
Anatomy(ANAT)	495	247
Chemical and Drugs(CHEM)	346	727
Devices(DEVI)	39	48
Disorders (DISO)	963	736
Geographic Areas (GEOG)	34	22
Living Beings (LIVB)	297	273
Objects (OBJC)	27	71
Phenomena (PHEN)	60	19
Physiology (PHYS)	160	119
Procedures (PROC)	574	433

2.2 Resources

Stanford Parser. As an existing open source toolkit, Stanford Parser is utilized to split sentences of the biomedical text. Furthermore, Stanford Parser also provides the function of POS tagging for multi-languages, such as English, Chinese, French, German and so on.

UMLS. Unified Medical Language System (UMLS) is used for mapping clinical entity to the unique concept identifiers (CUIs). And MetaMap[7] is a highly configurable application to map biomedical text to the UMLS metathesaurus or equivalently to identify metathesaurus concepts. This is the case of task 1b which is required to recognize clinical entities and their CUIs.

2.3 Feature Selection

Before model training, a large number of features need to be extracted from biomedical texts. The features can be categorized into four groups: lexical features, orthographic features, context features and lexicon features, listed in Tab. 3.

Lexical features use the first and the last four characters of token to identify the categories of entities. The POS of a token is helpful in named entity recognition. The

Stanford Parser tool is used to get POS tag of token, which is learnt on open domain corpus and supports multiple languages by loading template.

The tokens similar in shape can help the classifier "memorize" whether the token belong to one type of the entities. We replaced uppercase letters, lowercase letters, letters with diacritics and digits in a token by "A", "a", "b" and "0", respectively. Length of a token is a significant feature to clinical entity recognition. Similarly, information of capital letters is also a strong feature to help us identify the entities which always consist of uppercase letters. For example, the tokens of "Bio-safety Cabinet", "CT" and other proper noun can be identified by capital feature.

The context features of the classifier contain the lowercase, first four characters, last four characters, POS tags of two tokens before and after the current token.

Table 3. Features used in the CRF classifier.

Category	Feature
Lexical features	lowercase of the current token
	first four characters of the current token
	last four characters of the current token
	POS of the current token
Orthographic features	shape of the current token
	length of the current token
	whether the current token contains a letter
	whether the current token begins with a capital letter
	whether all characters in the current token are capital letters
	whether the current token contains a digit
	whether all characters in the current token are digits
	whether the current token consists of letters and digits
Context features	first four characters of two previous tokens
	first four characters of two next tokens
	last four characters of two previous tokens
	last four characters of two next tokens
	POS of two previous tokens
	POS of two next tokens
Lexicon feature	whether the current token is in the "GEOG" dictionary

Finally, a dictionary of geography based on French is extracted from webpage[8] of city, state and country. All the words in the dictionaries are lowercased. Lexicon features are used to judge whether the lowercase of the current token is in the dictionary or not, rather than as a feature of CRF model. If the current token shows up in the "GEOG" dictionary, we can conclude this token belongs to the entity of geographical category

After the features of token are generated, extracting an optimal subset from all the features is the most important step for building an effective classification model. At present, search algorithms can be divided into complete-based search, heuristic-based search and random-based search. The sequential forward selection (SFS) and sequential backward selection (SBS) based on heuristic are the most commonly-used

algorithms for selecting features. Beginning with an empty feature subset X, SFS add a feature x into X, and ensure the optimal performance of evaluation function J(X). After n-times iteration, the classification model is constructed based on local optimum. Instead of SFS, SBS starts a full feature set, and eliminate a feature from the feature set for each iteration.

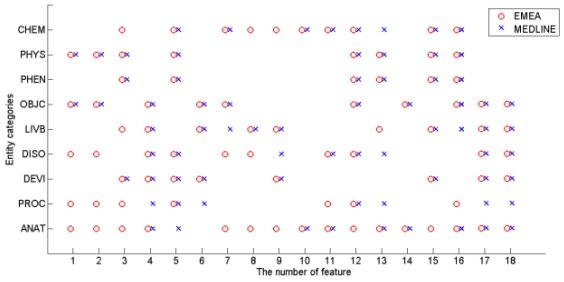


Fig. 1. The experiment is done to testify the effectiveness of BDS. The vertical and horizontal axes represent entity categories and feature categories, respectively. According to the different entity categories, WI-ENRE extracts the different feature set for building CRF model.

Compared with the above algorithms, we design and realize the bidirectional search (BDS) algorithm which combines the advantages of SFS and SBS, and improves the efficiency. The main idea of BDS is that SBS is used to search features, which is beginning with a full feature subset, while using SFS algorithm to search features beginning with an empty feature subset. Until a same feature subset is searched from both of SFS and SBS after n-iteration, BDS uses the same feature subset as the final results. For the feature selection of task 1b, the results for the different categories are shown in Fig. 1.

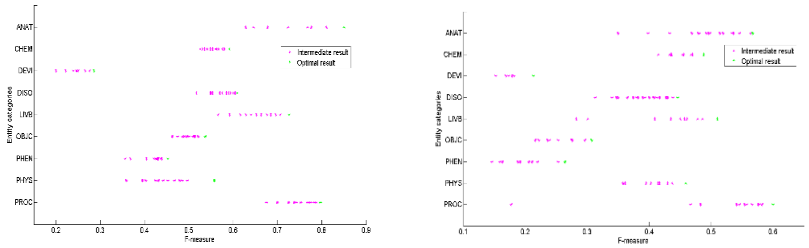


Fig. 2. The experiments of EMEA and MEDLINE demonstrate that the $F_{measure}$ of each categories change with the increase of iterations, and the most optimal combination of feature can be selected, respectively.

Furthermore, we list the $F_{measure}$ of the intermediate result, which is generated either SFS or SBS, in the process of n-iteration. For each category of entity, the most optimal combination of feature can be selected by BDS as shown in Fig. 2. Although

the method of feature selection may make out the local optimum, it can give better results than full feature subset for the feature selection of different entity categories.

2.4 Conditional Random Field

The conditional random field algorithm is proposed by Lafferty in 2001. CRF is arbitrary undirected graphical model that bring together the best of generative models and Maximum Entropy Markov Models (MEMM). A potential function is defined as follow:

$$\phi_{y_c}(y_c) = \exp(\sum_k \lambda_k f_k(c, y \mid c, x)) \tag{1}$$

Where $\phi_{y_c}(y_c)$ is a potential function of the fully connected network of Y, which is built on undirected graph. $y \mid c$ represents random variables which correspond to the cth node in the fully connected network by boolean form. Given an observed sequence of tokens, $x = x_1 x_2 ... x_n$, CRF can predicts a corresponding sequence of labels, $y^* = y_1 y_2 ... y_n$. y^*, which maximizes the conditional probability $p(y \mid x)$, is defined as follow:

$$p(y \mid x) = \frac{1}{Z(x)} \exp(\sum_{c \in C} \sum_k \lambda_k f_k(c, y_c, x)) \tag{2}$$

The conditional random field algorithm is widely used in named entity recognition. The existing open source toolkit CRF++[9] is utilized to classify the tokens in a sequence into the BIO scheme. The "B" indicates a token is the beginning of the clinical entity. The "I" represents that a token is inside of the clinical entity. The "O" means that a token does not belong to any category of the clinical entity.

Firstly, the training and testing data are generated based on the features. A CRF model can be learnt after training on the training data which is described in Sec. 2.1. Then the tokens in the testing data can be classified into one of the entity categories or non-entity category using CRF model.

3 Experiments

3.1 System Design

The WI-ENRE system consists of two main modules, ten sub modules and one evaluation module. The purpose of this system is to automatically identify clinically relevant entities in medical text in French.

- One of the major components is the named entity recognition module, which can identify the clinical entity based on Conditional Random Field and generate the specific model for each category. In the pre-processing, the biomedical texts are divided into two parts: MEDLINE and EMEA. Then, using the CRF model to recognize the clinical entity, the results will be evaluated and determined whether the feature set should be optimized. Until the results meet the optimization condition, the CRF models will be stored in the model repositories.
- The second module integrated with UMLS can select the CUIs to map clinical entity, and generate the annotated biomedical texts automatically. Besides English, UMLS does not support the other languages, such as French, Chinese and so on. Therefore, the API of Google is used to translate the entities from French to English in the first step. Then the translated entities are put into UMLS and mapped to the CUIs which is selected with the first result.

In the part of named entity recognition, the first step is the preprocessing of the file, which contains the part-of-speech tagging by Stanford Parser and the generation of training files based on entity category. The next step includes the training of CRF model, the decoding of CRF by testing files and the evaluation of entity results. Then the module of feature optimization is performed until the optimum result is found. Finally, all of the optimum model for each category will be stored into model repositories.

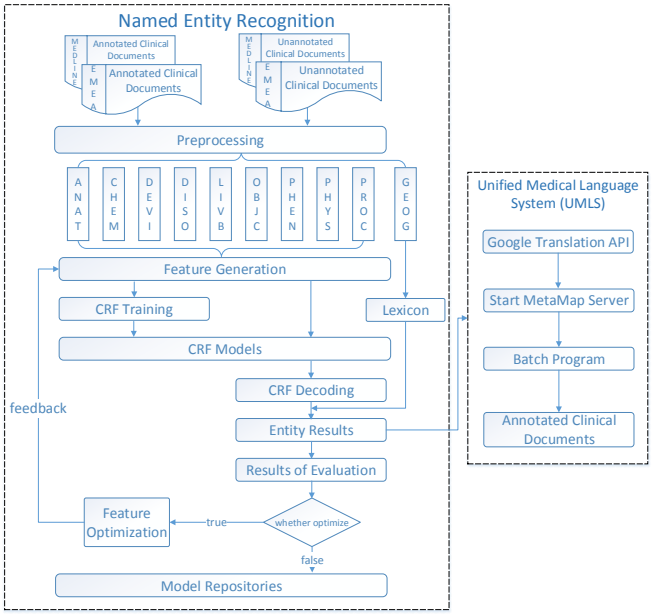


Fig. 3. The flow diagram of the WI-ENRE system is shown in this figure.

3.2 Evaluation Metrics

For task 1b, we determined the performance of WI-ENRE by comparing the system outputs against reference standard annotations. The system performance and performance for each category are evaluated rigorously. Precision, recall and $F_{measure}$[10] are calculated from true positive, false positive and false negative annotations, which are described as follows:

true positive (TP) = the annotation cue span from WI-ENRE overlapped with the annotation cue span from the reference standard

false positive (FP) = an annotation cue span from WI-ENRE did not exist in the reference standard annotations

false negative (FN) = an annotation cue span from the reference standard did not exist in WI-ENRE annotations

The formulas of the precision, recall, $F_{measure}$ are shown in Eqs. (3) - (5).

$$Precision = TP / (TP + FP) \tag{3}$$

$$Recall = TP / (TP + FN) \tag{4}$$

$$F_{measure} = 2 * Recall \bullet Precision / (Recall + Precision) \tag{5}$$

3.3 Recognition Accuracy

Using the evaluation metrics described above, the results of the WI-ENRE system are shown in Tab. 4 and Tab. 5.

Table 4. Results for each category/Phase 1 (EMEA):

	TP	FN	FP	Precision	Recall	$F_{measure}$
GEOG	22	7	3	0.880	0.759	0.815
DISO	225	233	141	0.615	0.491	0.546
LIVB	141	135	2	0.986	0.511	0.673
CHEM	183	687	18	0.910	0.210	0.342
OBJC	15	35	2	0.882	0.300	0.448
PHEN	4	6	6	0.400	0.400	0.400
PHYS	29	111	11	0.725	0.207	0.322
DEVI	2	20	3	0.400	0.091	0.148
ANAT	123	32	46	0.728	0.794	0.759
PROC	160	90	13	0.925	0.640	0.757
Exact match (official)	971	1289	234	0.429	0.805	0.56
Inexact match (official)	1137	1123	156	0.503	0.879	0.64

Table 5. Results for each category/Phase 1 (MEDLINE):

	TP	FN	FP	Precision	Recall	$F_{measure}$
GEOG	28	18	4	0.875	0.609	0.718
DISO	279	613	199	0.584	0.313	0.407
LIVB	142	178	28	0.835	0.444	0.580
CHEM	108	259	40	0.730	0.294	0.419
OBJC	8	27	10	0.444	0.229	0.302
PHEN	10	39	19	0.345	0.204	0.256
PHYS	31	120	53	0.369	0.205	0.264
DEVI	7	47	8	0.467	0.130	0.203
ANAT	232	262	78	0.748	0.470	0.577
PROC	267	302	188	0.587	0.469	0.521
Exact match (official)	1068	1909	671	0.358	0.614	0.452
Inexact match (official)	1523	1454	449	0.511	0.772	0.615

The evaluation results of EMEA and MEDLINE are presented respectively. The experiments show that results of EMEA are better than MEDLINE. In the 10 main categories, GEOG based on lexicon get the high $F_{measure}$ above 80 and 70 percent in different corpus. Compared to GEOG, the categories which are based on CRF, such as ANAT, PROC and LIVB, have a low $F_{measure}$ about 70 percent. In addition, the rest categories are worse than ANAT, PROC and LIVB, with below 50 percent. Through the analysis, it is observed that the entity categories of low accuracy do not basically select the orthographic features which are inside the feature range of 6th and 11th (as shown in Fig.1). Moreover, we also found that the entity categories which select the feature of POS get higher percentage of accuracy than others.

3.4 Error Analysis

The errors in the WI-ENRE system are analyzed according to the error analysis method[11], which is roughly divided into three groups: type error (entity is correct but type is wrong), missing error (entity is in the gold standard but not in the system output) and spurious error (entity is in the system output but not in the gold standard). Based on the types of errors, Tab. 6 lists the error distribution of WI-ENRE system.

Table 6. Error distribution of the WI-ENRE system at the clinical entity recognition of CLEFeHealth 2015 task 1b:

	Error number	Percentage
Type error	101	1.65%
Missing error	3,221	52.72%
Spurious error	872	14.27%

According to the three groups of error, missing errors make up the highest proportion as 52.72%. Therefore, the recall of the WI-ENRE system is very low.

Table 7. Error details of the WI-ENRE system at the clinical entity recognition of CLEFeHealth 2015 task 1b:

	System output											
	ANAT	CHEM	DEVI	DISO	GEOG	LIVB	OBJC	PHEN	PHYS	PROC	missing	total
ANAT		2		1		1			2		294	6
CHEM	2						1		1	1	946	5
DEVI	1									1	67	2
DISO	3	3	1			3			1	10	846	21
GEOG											25	0
LIVB	2			2					2		313	6
OBJC			3								62	3
PHEN				1					1	2	45	4
PHYS	2			17						2	231	21
PROC		1		22				4	2		392	29
Spurious	124	58	11	340	7	30	12	25	64	201		872
total	10	6	4	43	0	4	1	8	9	16	3,221	

The experiment shows that the categories of CHEM and DISO have high missing error with the count of 946 and 846, respectively. Twenty-two PROC entities are identified as DISO while 10 DISO entities are marked as PROC. It is difficult to distinguish between PROC and DISO for WI-ENRE. In addition, ANAT, LIVB, PHYS have a missing count of above 200. All of these led to the low recall rate of WI-ENRE system. Compare to missing errors, the spurious errors of DISO are also much higher than others. It follows that the system cannot recognize the category of DISO well, which not only has the higher missing errors but also is the most serious error of spurious. For the type error, a normal level which can be remained within acceptance criteria is shown in Tab. 7.

4 Conclusion

This paper described the clinical entity recognition by machine learning method for the task 1b of CLEFeHealth 2015. A suite of methods that included conditional random fields, feature selection with BDS algorithm and entity normalization using MetaMap performed the task well. Among these methods, the feature selection plays a crucial role to enhance the performance for each category. Using a suitable feature subset, we can obtain more accurate and reasonable classification than the full feature set. In order to testify this method, we design the system, WI-ENRE, to address the clinical entity based on CRF and achieve the normalization of clinical entity by UMLS.

The future study will be focused on the feature optimization and the improvement of recall rate. Moreover, the term vectors which are generated by word embedding can be taken as the characterizing attribute. The other useful features and more suitable methods will be researched to improve our system.

Acknowledgments. The MEDLINE title and EMEA documents used in this paper were provided by CLEFeHealth 2015 task 1b, and thanks to the organizing committee of CLEF and the annotators of the dataset.

References

1. Carol Friedman, Philip O. Alderson, John H. M. Austin, James J. Cimino, andStephen B. Johnson. A general natural-language text processor for clinical radiology. J Am Med Inform Assoc, 1(2):161-174, 1994.
2. Pierre Zweigenbaum. Menelas: an access system for medical records using naturallanguage. Computer Methods and Programs in Biomedicine, 45:117-120, 1994.
3. Ozlem Uzuner, Brett R. South, Shuying Shen, and Scott L. DuVall. 2010 i2b2/VAchallenge on concepts, assertions, and relations in clinical text. J Am Med InformAssoc, 18(5):552-556, Sep-Oct 2011. Epub 2011 Jun 16.
4. Hanna Suominen, Sanna Salantera, Sumithra Velupillai, Wendy W. Chapman,Guergana K. Savova, No_emie Elhadad, Sameer Pradhan, Brett R. South,Danielle L. Mowery, Gareth J. F. Jones, Johannes Leveling, Liadh Kelly, LorraineGoeuriot, David Martinez, and Guido Zuccon. Overview of the ShARe/CLEFeHealth evaluation lab 2013. In Proceedings of CLEF 2013, Lecture Notes in ComputerScience, Berlin Heidelberg, 2013. Springer.
5. Liadh Kelly, Lorraine Goeuriot, Gondy Leroy, Hanna Suominen, Tobias Schreck,Danielle L. Mowery, Sumithra Velupillai, Wendy W. Chapman, Guido Zuccon,and Joao Palotti. Overview of the ShARe/CLEFeHealth evaluation lab 2014. InProceedings of the ShARe/CLEFeHealth Evaluation Lab. Springer-Verlag, 2014.
6. Névéol A, Grouin C, Leixa J, et al. The Quaero french medical corpus: a ressource for medical entity recognition and normalization[C]//Proceedings of LREC BioTxtM 2014 Fourth Workshop on Building and Evaluating Resources for Health and Biomedical Text Processing. 2014.
7. Alan A. Aronson: Effective mapping of biomedical text to the UMLSMetathesaurus:the Metamap program. In: AMIA, pp. p.17-21. (2001).
8. http://www.culturecommunication.gouv.fr/.
9. CRFsuite package: http://www.chokkan.org/software/crfsuite/.
10. Hripcsak, G., Rothschild, A.: Agreement, the F-measure, and reliability in informationretrieval. J Am Med Inform Assoc 12(3) 296-8.
11. B. Wellner, M. Huyck, S. Mardis et al., “Rapidly retargetable approaches to de-identification in medical records,” Journal of the American Medical Informatics Association, vol. 14, no. 5, pp. 564-573, 2007.