2013 IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT)

Named Entity Disambiguation Using HMMs

Ayman Alhelbawy*†, Rob Gaizauskas*
*Sheffield University, Regent Court, 211 Portobello Street, Sheffield, S1 4DP, U.K
†Fayoum University, Fayoum, Egypt
ayman, R.Gaizauskas@dcs.shef.ac.uk

***Abstract*—In this paper we present a novel approach to disambiguate textual mentions of named entities against the Wikipedia knowledge base. The conditional dependencies between different named entities across Wikipedia are represented as a Markov network. In our approach, named entities are treated as hidden variables and textual mentions as observations. The number of states and observations is huge and naively using the Viterbi algorithm to find the hidden state sequence that emits the query observation sequence is computationally infeasible, given a state space of this size. Based on an observation that is specific to the disambiguation problem, we propose an approach that uses a tailored approximation to reduce the size of the state space, making the Viterbi algorithm feasible. Results show good improvement in disambiguation accuracy relative to the baseline approach and to some state-of-the-art approaches. Also, our approach shows how, with suitable approximations, HMMs can be used in such large-scale state space problems.**

I. Introduction

The relation between names and real world entities they denote is many-to-many: one entity may have several names; the same name may be shared by multiple entities. Establishing which real world entity a name mention denotes in a particular context is the problem of named entity disambiguation (NED).

In general, named entities (NEs) have a special importance in information extraction from text, or text mining. In the last two decades much research has been done on NE recognition and classification, and good progress has been made [1]. However, many recognized NEs are ambiguous and it is very important for software agents that aim to aggregate information on real world entities from sources such as the Web that they be able to identify which entities are the intended interpretation of different textual mentions. Also, it is important for search engines to correctly identify different names for the same named entity to get full coverage when searching for a named entity with different names.

The only available reference resources for real world entities are knowledge bases (KBs). Wikipedia is widely used as a reference KB to disambiguate ambiguous NE mentions by researchers working on this problem because of its breadth and free availability. It contains only references to relatively well known individuals, but nevertheless is suitable for research on this problem. We treat KB entries in Wikipedia as surrogates for real world entities. The textual portion of these entries typically contains mentions of multiple other NEs. When these mentions are hyper-linked to other KB entries we can infer that there is some relation between the real world entities corresponding to the KB entries. These links allow us to build up statistical co-occurence counts both between entities and between mentions and entities that occur in the same context

Thus, all the NEs in the KB can be represented in Markov network, where the nodes represent entities and the edges represent conditional dependencies computed between "grounded" (i.e. hyper-linked to a Wikipedia entry) mentions of the two connected entities and where the later occurrence is assumed to be dependent on the former. A Markov network is similar to Bayesian network in its representation of dependencies but differs in that Bayesian networks are directed and acyclic [2], while Markov networks need not be. For any text document containing a set of NE mentions, each NE mention may be mapped to a set of NEs in the Markov network, i.e. those entities which are possible interpretations of the name.

We treat the task of NED as finding the best sequence of KB NEs given a set of different mentions for different NEs in the same context. The KB named entity entry (state) is not directly visible and we have just the NE mention (observation). Thus, we employ an HMM approach and use Viterbi decoding to find the best NE entry (state) sequence that generates the mention (observation) sequence. HMMs have been successfully used for various sequence labelling tasks in language processing, including part-of-speech tagging [3], [4] and NE recognition [5]. But, to the best of our knowledge, such an approach has not been investigated for the NED task.

The huge number of states ($\sim 1M$) would appear to make it infeasible to use an HMM for this problem. However, we observe that by taking advantage of particular characteristics of the NE disambiguation task (e.g. considering only the set union of disambiguation sets for all NE textual mentions in a specific context of interest) we can reduce the state space, making the HMM approach feasible without affecting results.

II. Related Work

The task of Entity Linking (EL), as addressed in the TAC KBP challenges [6], is closely related to the NED task. However, the NED task is properly a subtask of the EL task because NED is concerned with disambiguating a textual NE mention, where the correct NE is known to be one of the KB entries, while the EL task also permits cases where there is no entry for the NE in the reference KB. NED may also be viewed as a limited form of word sense disambiguation [7], where only the names are to be disambiguated and the sense inventory is provided by the KB.

Many approaches have been used to tackle the NED problem. Initial *single entity approaches* used local context textual features from the query document. They compare these features to the textual features of NE descriptions in the KB and link to the most similar one [8]. A basic framework was defined by Bunescu and Pasca [9] to measure the similarity between the textual context of the NE mention and the Wikipedia

978-0-7695-5145-6/13 $31.00 © 2013 IEEE
DOI 10.1109/WI-IAT.2013.173

categories of the candidate. Cucerzan added more similarity features to this framework [10]. Some approaches deal with NED as a search result ranking problem. For example, Zheng et al. [11] and Dredze et al. [12] use supervised learn-to-rank models to re-rank the ambiguous candidate set. A document similarity function based on NEs is used in conjunction with learn-to-rank in a hybrid approach for disambiguation by Alhelbawy and Gaizauskas [13]. The main drawback of all of these approaches is that each NE is disambiguated independently of the others.

Collective disambiguation approaches have shown themselves to be more successful than single entity approaches. Milne and Witten [14] use a probabilistic graphical model to model the semantic relatedness between different NEs and use semantic relatedness between the candidate set and the unambiguous mentions in the context to disambiguate ambiguous NE mentions. Semantic relatedness is measured using Wikipedia incoming links for the candidate and unambiguous NEs. Another graph-based method for collective linking is presented by Hoffart et al. [15]. They build a graph of all textual mentions and all candidates for each mention. Then, the problem is formulated as finding the densest sub-graph that contains all textual mentions and one NE reference for each textual mention using three measures: the prior probability of the candidate to be mentioned, the textual similarity between the contexts of a mention and a candidate entity and the coherence among candidate entities. As finding the densest sub-graph is an NP-hard problem, they extend an approximation algorithm of [16] for the problem of finding strongly interconnected and size-limited groups in a social network. Kulkarni et al. [17] propose a collective approach that combines similarity of local context features to candidate entities with global topical coherence of candidate entity sets.

III. Framework

In this section, a detailed description of our problem formulation and approximation are presented. In our approach states are Wikipedia KB entries for NEs and observations are NE textual mentions or surface forms that appear in documents. Our proposed framework consists of two main modules. The first is an offline process to model all states and observations found in the Wikipedia KB. The second is a disambiguation module which uses the generated model to decode an input observation sequence into the most probable sequence of states that emits such an observation sequence.

A. Wikipedia HMM Modelling

We address only the most popular types of NEs: locations, organizations and persons. As a first step, the DBpedia ontology is used to build up a list of the Wikipedia NEs of types of interest, resulting in a list of 1,244,682 unique NE entries in the KB that are taken to be the KB reference states.

Next, Wikipedia[1] is parsed to identify all sequences containing one or more NE mentions (anchor text) linked, via an interwiki link, to one of the reference states. This yields a list 1,629,961 of NE textual mentions extracted from the Wikipedia anchors. This list is used to form the observation dictionary. Dependencies between different NEs in the Wikipedia KB and emission probabilities for different textual mentions are modelled using the HMM trainer.

To model NE dependencies we explore four different assumptions about the textual scope, or "context" of NE dependencies, defining a segmentation scheme for each:

sent: NE context is a single sentence.
par: NE context is a single paragraph.
seg: NE context is a Wikipedia article subsection.
doc: NE context is the entire document.

HMM models were trained using NE sequences extracted from the Wikipedia KB according to the different segmentation schemes. Thus, different models are trained with the same NEs but using different sequences.

Because of the huge number of extracted sequences, training is slow[2] and the resulting models are very large. To alleviate these issues we reduced the number of sequences used in training by considering only sequences that have at least one of the test mentions or test named entities (of course, many named entities or states other than those used in test data will still appear in these sequences). Overall this reduced the number of sequences used in training by 30-70% and the number of states in the resulting models by 17-60%, depending on the sequence type. For each context scheme a model μ is generated, yielding four models: μ_{sent}, μ_{par}, μ_{seg}, and μ_{doc}.

B. HMM disambiguation

The proposed formulation for NED takes the form of a hidden Markov model (HMM), where the states correspond to the Wikipedia NEs E and mentions M are emitted each time a state is visited. HMM NED training involves estimating the different mentions emission probabilities $p(m_i|e_j)$ and the transition probabilities between different states $p(e_i|e_{i-1})$. Given an observation sequence $M_{test} = m_1, \ldots, m_n$ where $n \geq 1$ the goal of NED is to find a stochastic optimal tag sequence $E_{test} = e_1, \ldots, e_n$ that maximizes the joint probability of a sequence of states and observations $Pr(e_{1:n}, m_{1:n})$, where $x_{1:n}$ abbreviates $x_1, \ldots, x_n$. According to the first order HMM assumption, this joint probability is given by:

$$Pr(e_{1:n}, m_{1:n}) \approx p(e_1)p(m_1|e_1)\prod_{i=2}^{n} p(e_i|e_{i-1})p(m_i|e_i) \quad (1)$$

The Viterbi algorithm is a well known algorithm that is used to find the most probable state sequence that emits an unlabelled observation sequence [18]. We used Viterbi algorithm to find the best NE sequence that emits the unlabelled sequence of mentions. However, using Viterbi algorithm with all states can be computationally infeasible when working with a very large-scale state space. Suppose N is the number of NEs found in Wikipedia and M is the number of observations/textual mentions found in Wikipedia. $N \approx 10^6$ and $M \approx 1.5 \times 10^6$. Then, the size of transition matrix A that keeps the transition probabilities between all states is $N^2 \approx 10^{12}$. The size of the emission probability matrix B that stores for each state the probability of that state emitting one of the observations is $N \times M \approx 1.5 \times 10^{12}$. Despite the sparseness of both matrices, all states must be visited to calculate the best path according to the Viterbi algorithm. The complexity of the Viterbi algorithm is $O(T \times N^2)$ where T is the sequence length.

[1] We use a Wikipedia dump taken in February 2012.

[2] For the reduced datasets described below, training took about 5 days/model.

For the NED problem, where for each mention the correct entity is assumed to be a known entry in the KB, the set of candidate disambiguation states for each observation is known or can be constructed using simple techniques. Therefore it is a waste of time to try to find the best sequence from amongst *all* states when the correct one is partially known given the disambiguation sets. Thus, we observe that for any ambiguous observation M_i there is a set of candidate states $E_i = \{e_i^1, \ldots, e_i^j\}$, where j is the number of disambiguation candidates. Typically, $|E_i| \ll |E|$.

Assuming the correct NE falls into the NE candidate list of each mention, a new state space of the disambiguation set of NEs for all mentions in the sequence can be used instead of using the original state space that includes all states. This approximation makes it computationally feasible to find the best sequence using the Viterbi algorithm. We propose three variants approximations for reducing the state space to decode a textual NE mention sequence into a sequence of Wikipedia knowledge base NEs. We have adapted the Viterbi algorithm to handle these approximations.

Approximation (1): Suppose we have a dependency model μ that models the dependency network of the states E. In this approximation, the state space is reduced to include only the set of states that emits any of the observations in the input sequence M, where $M = m_1, m_2, ...m_n$. Hence a new state space E_{seq} is defined where $E_{seq} \subset E$ and $E_{seq} = \{E_1 \bigcup E_2 \bigcup ... \bigcup E_n\}$ with $E_i = \{e_i^1, e_i^2, e_i^3, ..., e_i^k\}$, k being the number of NE candidates for a mention m_i and $1 \leq i \leq n$. Then, $E_{seq} = \{e_{seq}^1, e_{seq}^2, e_{seq}^3, ..., e_{seq}^c\}$, where c is the total number of all candidates in all lists. The HMM decoding steps, as adapted from Manning and Schütze [18], are as follows:

1) Initialization:

$$\delta(j) = p(e_{seq}^j) \times p(m_1|e_{seq}^j)\ ,\ 1 \leq |E_{seq}|$$

2) Recursion:

$$\delta_t(j) = max_{1 \leq i \leq c} \left[\delta_{t-1}(i) \times a_{ij}\right] \times p(m_t|e_{seq}^j)$$

where $1 \leq j \leq c\ ; 2 \leq t \leq k\ and\ c = |E_{seq}|$

Note here that states considered are just those in E_{seq} rather than in E.

Approximation (2): In approximation (1) the state space is reduced to include only those states, i.e. entities, which are deemed potential states to emit the observations, i.e. mentions, in the test sequence to be labelled. However, a further observation is that not all states in this restricted state space are possible states for all observations. In approximation (2) not only is the state space customized for each test sequence to be labelled; the state space is determined for each NE textual mention. I.e. in this approximation, we reduce the state space for each mention separately. So, for every NE mention there is a specific state list. Hence a new E_{seq} will be defined where $E_{seq} \subset E$ and $E_{seq} = \{E_1, E_2, ...E_n\}$ and $E_i = \{e_i^1, e_i^2, e_i^3, ..., e_i^k\}$ where k is the number of NE candidates for a mention m_i. The HMM decoding steps in this case are:

1) Initialization:

$$\delta(j) = p(e_1^j) \times p(m_1|e_1^j)\ ,\ 1 \leq j \leq |E_1|$$

2) Recursion

$$\delta_t(j) = max_{1 \leq i \leq |E_t|} \left[\delta_{t-1}(i) \times a_{ij}\right] \times p(m_t|e_j^t)$$

where $1 \leq j \leq |E_t|\ ; 2 \leq t \leq k$

Note that in this case a separate set of states is considered for each mention.

Approximation (3): In collective NED approaches a basic intuition is that NEs mutually inform the disambiguation of each other. This mutual information is inherently order free, i.e. independent of the order in which the NEs appear in the training or test sequences.

In this variant of our approach, the state space is reduced for each textual mention separately as in approximation (2), but with one difference which is reordering the mention sequence by ambiguity degree, i.e. by the number of candidate NEs for each textual mention, from lowest to highest. HMM decoding steps are the same as in approximation (2). However, we now compute the state sequence probability for the query mentions twice: once in the order they occur in the query document and once after reordering the query mentions according to ambiguity degree. The state sequence with the highest probability is considered the solution for the query sequence.

IV. Dataset

There is no established benchmark for NED. The only available related benchmark is the TAC KBP entity linking task. As discussed in Section II above, the EL task is different from the NED task and furthermore in the TAC KBP benchmark only one NE textual mention is to be disambiguated in the query document, while for NED all mentions must be disambiguated. Thus the KBP data is not suitable for evaluating collective NED approaches. There are hand-annotated datasets for NED like that reported in [17], but this is quite small and uses an old version of Wikipedia. We used the AIDA dataset, a dataset based on the CoNLL 2003 data for NER tagging in which most tagged NE mentions have been disambiguated against Wikipedia [15]. The dataset contains 1,393 documents with 34,956 NE textual mentions of which 27,817 have been disambiguated against Wikipedia.

As discussed on HMM modelling, DBpedia is used to get a list of all NEs found in Wikipedia, where the NE must be of type Person, Location or Organization. However, not all NEs in Wikipedia are classified in DBpedia. In particular, some NEs that occur in AIDA are not included in DBPedia. Since this results in no training sequences that mention these NEs being extracted from Wikipedia, we added all NEs in AIDA not already in the DBpedia NE set to this set.

V. Experimental Results and Discussion

Our baseline is a simple statistical approach that uses co-occurrence counts of NE mentions and the NEs to which they are linked in Wikipedia. The NE mention is always disambiguated to the NE with which it is most frequently associated in the Wikipedia pages.

During testing, the test data is segmented using three context schemes: sent, par and doc. While the test collection is not segmented into paragraphs, a heuristic is used to build

Test contxt	μ	Approx1 Accuracy		Approx2 Accuracy		Approx3 Accuracy	
		macro	micro	macro	micro	macro	micro
sent	μ_{sent}	62.09	69.46	69.33	74.65	73.76	78.09
sent	μ_{par}	62.44	69.76	69.72	74.97	73.96	78.18
sent	μ_{seg}	62.1	69.91	70.24	**75.55**	**74.18**	**78.49**
sent	μ_{doc}	60.06	68.74	70.03	75.50	73.90	78.13
par	μ_{sent}	55.12	61.07	68.98	72.06	73.54	75.89
par	μ_{par}	55.87	61.87	69.17	72.42	73.90	76.39
par	μ_{seg}	56.79	63.09	70.14	73.51	73.99	76.94
par	μ_{doc}	56.1	62.80	**70.28**	73.76	74.00	77.09
doc	μ_{sent}	50.32	57.01	68.69	71.27	69.64	72.86
doc	μ_{par}	50.90	57.78	68.60	71.50	69.55	73.27
doc	μ_{seg}	51.82	59.12	68.92	72.15	69.66	73.73
doc	μ_{doc}	52.4	59.47	69.08	72.34	69.74	73.84

TABLE I. RESULTS OF USING DIFFERENT APPROXIMATIONS FOR HMM DISAMBIGUATION

	approx1	approx2	approx3	Baseline	Cucerzan	Kulkarni	Hoffart
A_{macro}	62.10	70.24	74.18	44.06	43.74	76.74	81.91
A_{micro}	69.91	75.55	78.49	43.55	51.03	72.87	81.82

TABLE II. HMM AND STATE-OF-THE-ART RESULTS

up artificial paragraphs by taking consecutive sentences until the number of NE mentions is 3 or more. For each context scheme, the NE textual mention sequence is extracted and the HMM is used to label this mention sequence.

A set of experiments was carried out to test the accuracy of different approximations using the different models and different contexts. Micro and macro accuracy are used as the evaluation measures. Micro accuracy (A_{micro}) is accuracy over all NE mentions in the test set, while macro accuracy (A_{macro}) is the accuracy per document averaged over all test documents. Results are presented in Table I and show best accuracy occurs when using Approximation 3, sentence context in the query and the model μ_{seg}. For each approximation, the query context segmentation scheme has an impact on the results, while the effect of changing models is very slight. Theoretically, using shorter model contexts may divide the longer sequences into a smaller sequences. Then, transition probabilities of states at sequence boundaries are affected.

It is difficult to compare our results to the state-of-the-art results because there is no standard benchmark. Hoffart et al. [15] re-implemented the methods of Cucerzan [10] and Kulkarni [17] and evaluated them using the AIDA dataset. Table (II) shows a comparison between the results of our approach, the baseline approach and some state-of-the-art results. Our approach exceeds the results of Cucerzan and Kulkarni but does not exceed Hoffart's results. However our approach is very simple and direct to apply, unlike Hoffart's which is considerably more complex than ours.

VI. CONCLUSIONS AND FUTURE WORK

Our results show HMMs can be used as an effective approach to collectively disambiguate different textual mentions of different NEs in a document. Our proposed approximations make using HMMs feasible for the NED task, where both numbers of states and observations are huge. Using the information contained in the joint presence of NEs may not be sufficient to solve the NED problem on its own, but it goes a surprising way towards doing so.

In future work we plan to retrain our models using all of the Wikipedia data. Another possible extension is to adapt the measure used for emission probability to include a measure of candidate certainty based on context similarity.

REFERENCES

[1] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[2] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, ser. Adaptive Computation and Machine Learning Series. Mit Press, 2009.

[3] R. Garside, "The CLAWS Word-tagging System," in: R. Garside, G. Leech and G. Sampson (eds), *The Computational Analysis of English: A Corpus-based Approach*. London: Longman, 1987.

[4] K. Church, "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text, In *Proceedings of the Second Conference on Applied Natural Language Processing*, Austin , Texas , 1988, pp. 136-143.

[5] D.M. Bikel, S. Miller, R. Schwartz and R. Weischedel, "Nymble: a high-performance learning name-finder," in *Proceedings of the fifth conference on Applied natural language processing*, 1997, pp. 194–201.

[6] P. McNamee and H. Dang, "Overview of the tac 2009 knowledge base population track," in *Text Analysis Conference (TAC)*, 2009.

[7] R. Navigli, "Word sense disambiguation: a survey," ACM Computing Surveys, vol. 41, no. 2, pp. 1–69, 2009.

[8] R. Mihalcea and A. Csomai, "Wikify!: linking documents to encyclopedic knowledge," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007.

[9] R. C. Bunescu and M. Pasca, "Using encyclopedic knowledge for named entity disambiguation," in *Proceedings of the 11th Conference of the European Chapter of the ACL (EACL)*, 2006.

[10] S. Cucerzan, "Large-scale named entity disambiguation based on wikipedia data," in *Proceedings of EMNLP-CoNLL*, vol. 6, 2007.

[11] Z. Zheng, F. Li, M. Huang, and X. Zhu, "Learning to link entities with knowledge base," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, ser. HLT '10, 2010, pp. 483–491.

[12] M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin, "Entity Disambiguation for Knowledge Base Population," in *Proceedings of the AAAI Spring Symposium on Learning by Reading and Learning to Read*, vol. 3, 2009, pp. 76–82.

[13] A. Alhelbawy and R. Gaizauskas, "Named entity based document similarity with svm-based re-ranking for entity linking," in *Advanced Machine Learning Technologies and Applications*. Springer Berlin Heidelberg, 2012, vol. 322, pp. 379–388.

[14] D. Milne and I. Witten, "Learning to link with wikipedia," in *Proceeding of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 509–518.

[15] J. Hoffart, M. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum, "Robust disambiguation of named entities in text," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 782–792.

[16] M. Sozio and A. Gionis, "The community-search problem and how to plan a successful cocktail party," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 939–948.

[17] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti, "Collective annotation of wikipedia entities in web text," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 457–466.

[18] C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.