ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Structured Information Retrieval of Natural Language Supporting Clinical Decision-making

Bachelor Thesis

Kevin Klein

March 7, 2017

Advisors: Prof. Dr. Thomas Hofmann, Dr. Carsten Eickhoff
Department of Computer Science, ETH Zürich

Abstract

The aim of this thesis lies in enhancing medical information retrieval by comparing documents based on their affiliation to human-understandable clinical concepts instead of mere exact term-matchings. By doing so, we hope to bridge the gap in information retrieval quality between exact term-matching and semantic matching, as perceived by humans. In particular, its application should allow for reliable recommendation of biomedical literature based on digital natural language patient records and thereby facilitating the life of decision-making clinical personnel. In order to improve retrieval quality by overcoming vocabulary mismatches, our method seeks to combine BM25 term-frequency information retrieval with explicit topic modeling similarities based on ICD-9 codes, representing findings. Upon comparing our results to a baseline of pure BM25 or previous latent topic modeling attempts, our method underperforms with respect to multiple precision metrics. Further investigation into UMLS concept extraction indicate that explicit topic modeling with specific medical diagnoses and symptoms is not advisable for this task.

Acknowledgements

I wish to thank Dr. Carsten Eickhoff for patiently guiding me, Prof. Thomas Hofmann for allowing me to write the thesis in his research team as well as ETH IT-Services for providing access to the Euler compute cluster.

Contents

Chapter 1

Introduction

Recent findings indicate that the growth rate of scientific publications is exponential [9]. Hence, in addition to a substantial, already existing corpus of literature, it costs physicians of diverse specializations a tremendous effort to search for information. Moreover, it follows directly that consciously remembering even contextual subsets of this abundance is not demandable from a human. In other words, proposing a caretaker a small selection of highly relevant information based on already captured knowledge would turn diagnosing and choosing a treatment or examination into a more effective and more efficient process. Considering the liabilities involved in all clinical decisions, transparency is a key factor with respect to a method's potential for real-world use.

Current approaches to solving this clinical decision support task typically incorporate exact matchings of words in patient reports and scientific articles, elaborate techniques hiding the rationale for relevance, such as latent topic models, or both. The issue with the former is that lacking generality leads to reduced retrieval potential. As patient reports neither support well-defined formats nor a standardized vocabulary, small variations in the wording of patient notes will make some present concepts undiscoverable and unretrievable [2]. Studies have shown that the likelihood of specialists selecting equal terminology to describe the same subject is as low as 20% [4] and that a query term fails to appear in 30-50% of relevant documents [14] in standard IR scenarios. A disadvantage of the latter option is that they are highly dependent on the dataset and might therefore not be useful for other datasets in general. Another substantial downside is that transparency, comprehensibility and reproducibility are of great importance in decision processes where technology complements humans, as is the case in our scenario. A ”blackbox” carries a serious risk of suffering poor adoption for both legal and personal reasons.

Among the possible choices for explicit topic models, we narrowed our

choice down to pre-existing controlled vocabularies or thesauri. Firstly neither tools nor labeled datasets for manually crafted concepts exist. Secondly, we did not encounter any terminology unable to be expressed by one of the well-developed concept identification sets such as ICD-9, UMLS or MeSH[1]. Consequently we have both looked into ICD-9 codes and UMLS concept unique identifiers (CUIs). The MIMIC database[2] offered a very large number of patient reports, labeled with ICD-9 codes, which we were able to train binary classifiers on. QuickUMLS is a tool for concept extraction we used to obtain UMLS CUIs.

The thesis envisions to create a composite retrieval system based on both BM25 and explicit topic modeling. After all, exact-matching BM25 provides a solid baseline for the retrieval of a very high variety of data points. In order to substitute BM25, explicit topic modeling would have to encompass a high amount of dimensions and thereby risk to lose its expressiveness. Therefore we aimed to combine both methods and to determine each method's weight empirically.

In Chapter 2, we will guide through relevant work with respect to previous attempts at solving the TREC Decision Support Challenge as well as established topic modeling archetypes. Chapter 3 will familiarize the reader with the methods and concepts used to construct and evaluate our system. Information to the utilized data, heuristics, intermediary decisions and results will be made accessible in Chapter 4, before discussing the results in Chapter 5 and drawing the conclusions in Chapter 6.

[1] https://www.nlm.nih.gov/mesh/
[2] https://mimic.physionet.org/

Chapter 2

Related Work

2.1 Previous TREC CDS attempts

2.1.1 Word-embeddings, negation tagging and latent topic model

Last year, a submission from ETH's Data Analytics Lab has been able to perform very well by combining three successive methods [6]. In order to provide greater context, queries have been expanded by literal matchings with MeSH concepts as well as word2vec word-embeddings. Consequently, new term representations for negations have been introduced in order to explicitly match negations in both queries and documents. Lastly, the scores underwent a re-ranking based on a latent dirichlet allocation over latent topic models to account for variations in query and document vocabularies.

2.1.2 Explicit and latent topic models

Balaneshinkordan, Saeid and Kotov focused on combining both explicit and latent topic models [1]. Their idea revolves around determining the relative importance of explicit and latent models by optimizing weights. The basis of retrieval were markov random fields and the range of concepts covered extracted UMLS CUIs, google search results and pseudo relevance feedback documents. By leveraging both semantic and statistical concepts from various sources, they claim to have outperformed the median submitted run by 70 to 86%.

2.1.3 Memory network and knowledge graph

Hasan et al. achieved better than median performance in over 40% of the queries by merging a deep-learning and knowledge driven approach [8]. After building a knowledge graph based on wikipedia's clinical medicine categories, a query could be located by its underlying context. Furthermore,

Table 2.1: Categorization of topic modeling techniques

	Probabilistic	Non-Probabilistic
Explicit	EDA, LDA-STDW	ESA
Latent	LDA	LSA

their novel usage of key-value memory networks for diagnostic inferencing proved useful.

2.1.4 Multitude of term matchings and position-aware classification

In 2015, Song, He, Hu and He expanded the queries by MeSH concepts extracted from google search results [12]. In order to make use of different retrieval systems, they simultaneously used scores from BM25, PL2 and BB2 retrieval models. The 3-dimensional score vectors then underwent a re-ranking via either pointwise Random Forest or pairwise SVM classifiers while taking the positions of terms within a text into account.

2.2 Topic modeling in information retrieval

The realm of topic modeling being broad and diverse, we will try to highlight key differences by distinguishing approaches with respect to two dimensions: explicit/latent and probabilistic/non-probabilistic. We give an example for each tuple and illustrate their orientation in Table 2.1.

2.2.1 Explicit Semantic Analysis

Explicit Semantic Analysis (ESA) describes the meaning of a text by its relatedness to concepts, modeled by wikipedia articles [5]. The method is based on a Semantic Interpreter consisting of a weighted inverted index. This index allows for each term of the input text to be associated with wikipedia articles and their respective weight. Semantic interpretation of all words of the input text then yields an overall weight for each article associated with at least one term. The input text is represented by its interpretation vector, holding the association values for articles contained in a subset of user-defined concepts. Such interpretation vectors can then be used to compare different texts by using e.g. Cosine Similarity.

2.2.2 Latent Semantic Analysis

Latent Semantic Analysis (LSA) of relations between different bodies of text is based on an occurrence matrix. Its columns represent the inspected bodies of text with rows corresponding to the terms. Typically, with a vocabulary much larger than the size of each body of text, the matrices have a sparse

shape. The key idea is to reduce the dimensionality of the text corpora by transforming the matrix to a form with only ”few” rows. This step consists of applying Singular Value Decomposition (SVD) to the occurrence matrix. According to the properties of SVD, the applicant can choose a suitable n and represent the data by the n components associated with the n singular values with greatest magnitude and obtain a rank-n approximation with minimal error in Frobenius norm. In the context of our application, we must note that the interpretation of resulting dimensions might be hindered, as each can be a combination of several terms. In addition, polysemy, i.e. multiple meanings of a word, cannot be expressed in LSA.

2.2.3 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) ”discovers” hidden topics by formulating a generative model for documents [3]. The number of topics to be inferred can be chosen as a parameter. By assuming that each word in a document has been generated by one of the document's topics, a full probability distribution is learnt iteratively. Hence, a document can be represented as fractions of topics. In short, the latent topics can be interpreted by looking into which terms they are highly associated with and can therefore be thought of as groups of related words. E.g. the topic ”computer-related” is likely to induce high probabilities for the generation of ”keyboard”, ”memory” and ”algorithm”, while more general terms such as ”the” and ”high” will be assigned lower probabilities.

2.2.4 Probabilistic Explicit Topic Modeling

Hansen et al. [7] proposed to combine explicit topics with a probabilistic methodology. Their methods, EDA and LDA-STWD, use term distributions from wikipedia as approximate a priori topic-word distributions. In particular, it was their aim to address the lack of identifiability of latent topics. Despite underperforming to explicit non-probabilistic topic models in dimensionality reduction for document classification, their methods have an edge most techniques in document label generation.

Given the presented approaches, our method is conceptually closest to ESA. As our corpus consists of very specialized content, using a subset of wikipedia articles as topics seems too general, especially as the terms appearing in wikipedia articles tend to be understandable for laymen whereas the vocabulary of patient notes is heavily specialized. Moreover, learning on the already existing classification labels provided much greater potential than labeling documents in an unsupervised fashion.

Chapter 3

Methodology

3.1 Ranking functions

In the following we will present two very popular functions assigning a similarity score to a query-document pair. Both models rely on exact matchings of terms and assume a bag-of-words model, i.e. the order of words is in a document is ignored, yet their respective count is taken into consideration.

3.1.1 Notations

We will first introduce some notations to simplify upcoming formulas.

- D: corpus of documents
- d: document from corpus D
- t: term or word from a document
- N: number of documents in corpus D
- N_t: number of documents in corpus D containing term t
- L_d: length or amount of terms in document d
- $L_{avg,D}$: average length of documents in corpus D
- Q: query consisting of the terms $q_1, \ldots, q_n$

3.1.2 Tf-idf

Tf-idf is a ranking function expressing the importance of a term to a specific document of a corpus. This can be achieved via adjusting the term frequency in a document by its frequency in the whole corpus. The exact formulas can vary based on the choice of normalization. The "raw" frequency of t in d is

computed as follows.

$$f_{t,d} = \frac{\text{\# occurrences of } t \text{ in } d}{\text{\#terms in } d}$$

Thus, with an exemplary form of normalization we obtain:

$$tf(t,d) = 1 + log(f_{t,d})$$

The inverse document frequency can vary regarding normalization as well. We introduce one possible form.

$$idf(t,d) = log\frac{N}{N_t}$$

Thereby the score of Q and d is understood as:

$$score(Q,d) = \sum_{i=1}^{n} tf(q_i,d) \cdot ift(q_i,d)$$

3.1.3 BM25

BM25, also known as Okapi BM25 is a popular ranking function introduced at TREC-3 [10]. Similarly to tf-idf, it assigns scores to documents based on the frequencies of exactly matching terms. BM25 uses a distinct variation of idf normalization.

$$idf(t,d) = log\frac{N - N_t + 0.5}{N_t + 0.5}$$

This leads to the following score function:

$$score(Q,d) = \sum_{i=1}^{N} idf(q_i,d) \cdot \frac{f_{q_i,d} \cdot (k_1 + 1)}{f_{q_i,d} + k_1(1 - b + b\frac{L_d}{L_{avg,D}})}$$

The parameters b and k_1 can be used for optimization. A standard choice is $b = 0.5,\ k_1 = 1.2$, as is default in the Apache Lucene[1] search engine.

3.2 Binary classifiers

We will introduce two very common probabilistic classifiers. Learning from a set of labeled training data, they will be used to predict labels, i.e. classes, for unlabeled data. In other words, we seek a probability distribution $\Pr(c_i|d)$ where $c_i \in \{c_1, c_2\}$ is one of both classes. If we are given such a distribution, we can simply pick the class c_i maximizing this value. Notations will be inherited from the previous subchapter.

[1] https://lucene.apache.org/

3.2.1 Naive Bayes

The Naive Bayes classifier is based on the assumption that the individual terms of an input document are conditionally independent. This assumption rarely holds true, yet the lack of accuracy is traded off for convenient simplification of the calculations.

$$\Pr(d|c_i) = \prod_{t \in d} \Pr(t|c_i)^{f_{t,d}}$$

In addition, the model applies the following heuristics on the training set to define the following probability distributions.

$$\Pr(c_i) = \frac{\#\ d \in D \text{ s.t. class of } d \text{ is } c_i}{N}$$

$$\Pr(t|c_i) = \frac{\sum_{d \in D} tf(t,d)}{\sum_{d \in D} L_d}$$

Unlabeled data can now be classified with a so-called "maximum a posteriori" rule.

$$\begin{aligned} c(d) &= argmax_{c_i \in \{c_1, c_2\}}\ \Pr(c|d) && (3.1)\\ &= argmax_{c_i \in \{c_1, c_2\}}\ \frac{\Pr(d|c_i)\Pr(c_i)}{\Pr(d)} && (3.2)\\ &= argmax_{c_i \in \{c_1, c_2\}}\ \Pr(d|c_i)\Pr(c_i) && (3.3)\\ &= argmax_{c_i \in \{c_1, c_2\}}\ (\prod_{t \in D} \Pr(t|c_i)^{f_{t,d}})\Pr(c_i) && (3.4) \end{aligned}$$

3.2.2 Logistic Regression

As opposed to fitting distributions for each feature of a datapoint, Logistic Regression (LR) fits all weights together. The key difference to Naive Bayes is that LR takes correlation of features into consideration instead of assuming their independence. Let us define the function $f(d) = w \cdot h(d)$ where w is a weight vector and $h(d)$ a feature-vector of d. In our case, $h(d)$ represents the vector of term frequencies over the vocabulary of the corpus. Intuitively, we could create a "hard" classifier such that $\Pr(c_i = c_1|d) = 1$ if $f(x) > 0$. However, this implied that the misclassification of a single datapoint would yield a likelihood of 0 for the dataset. Hence, we are required to model "noise" and will assume that $\Pr(c_i|d)$ can be modeled as a Gaussian. Furthermore, we want it to be symmetric and both classes to be equally likely for $f(d) = 0$. Taking normalization into account, we obtain the following.

$$\Pr(c_i = c_1|d) = \frac{exp(w \cdot h(d))}{exp(w \cdot h(d)) + 1}$$

$$\Pr(c_i = c_2|d) = \frac{1}{exp(w \cdot h(d)) + 1}$$

In order to determine the optimal weight-vector w, we define the log-likelihood.

$$\mathbb{L}(w) = log(\prod_{d \in D} \Pr(c(d)|d, w) = \sum_{d \in D} log \Pr(c(d)|d, w)$$

This quantity can be optimized through its gradient, i.e. its derivative with respect to w. Unfortunately, there is no analytical closed form for w by setting the gradient equal to 0. Nevertheless, observing the convexity of this function allows us to state that the log-likelihood for an initial guess w_0 increases upon moving into the direction of the gradient.

$$\mathbb{L}(w_0 + \alpha \nabla \mathbb{L}(w_0)) \geq \mathbb{L}(w_0)$$

Using this knowledge, we can construct the iterative procedure called *gradient ascend* to converge to the optimum. The latter is guaranteed to be unique thanks to convexity. We call α the learning rate.

$$w^{k+1} := w^k + \alpha \nabla \mathbb{L}(w^k)$$

3.2.3 Evaluation: accuracy

Accuracy is a metric to describe the performance of a classifier.

$$accuracy = \frac{\#\text{True negatives } + \#\text{True positives}}{\#\text{Negatives} + \#\text{Positives}}$$

3.3 Cosine similarity

In order to operate on queries and documents, they are typically transformed to and represented in a well-defined feature-space – just as document d was utilized in the shape of $h(d)$ in the previous subsection. It is important to note that this transformation is equally applied to all inputs and therefore their projections populate the same vector space. It follows that their projections are of equal dimension. In order to compare the similarity of a pair of m-dimensional vectors x and y, we take advantage of this metric, operating on the angle $\theta(x, y)$ between both vectors.

$$sim(x, y) = cos(\theta(x, y)) = \frac{x \cdot y}{\|x\| \|y\|} = \frac{\sum_{i=1}^{m} x_i y_i}{\sqrt{\sum_{i=1}^{m} x_i^2} \sqrt{\sum_{i=1}^{m} y_i^2}}$$

3.4 Normalization: rescaling

In order to combine scores from different ranges, we introduced a normalization scaling to a common range of results, namely $[0, 1]$. Hereby we denote

$\{x_i\}$ to be the set of values to be normalized and x'_k the normalized value of $x_k \in \{x_i\}$.

$$x'_k = \frac{x_k - min\{x_i\}}{max\{x_i\} - min\{x_i\}}$$

3.5 Overview of composite retrieval

The key idea of our method is to merge exact term-matching information retrieval via BM25 and concept-matching classification.

Given the set of documents and queries – $[d]$ and $[q]$ in Figure 3.1 – the application of BM25 returns a ranking indicating a score s per query-document pair. This score s is the foundation of our retrieval. In addition, each of the m binary classifiers predicts document and query labels individually, yielding a probability vector p_q and p_d each. Subsequently, the cosine similarity of m-dimensional probability vectors is computed for each query-document pair. The final result is then obtained by synthesising both intermediary values, normalized by rescaling. Per query-document pair we compute the final score s'.

$$s' = (1 - \lambda) \cdot norm(s(q, d)) + \lambda \cdot norm(cosSim(p_q, p_d))$$

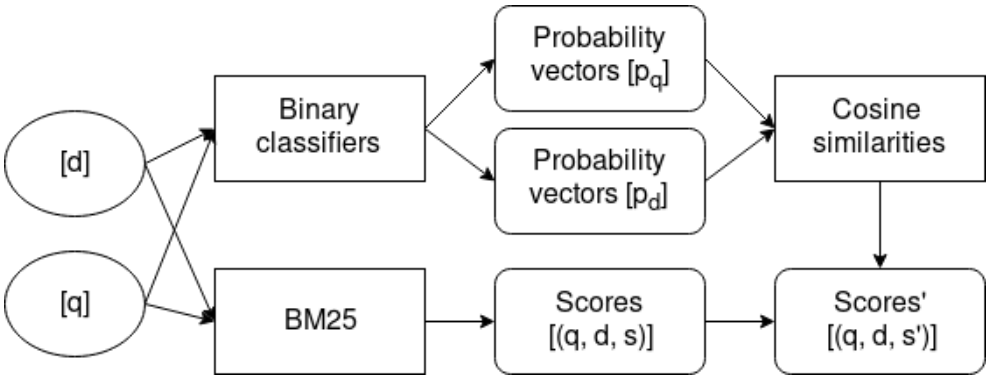

Figure 3.1: Overview of our composite retrieval system

3.6 Information retrieval performance metrics

In order to illustrate the most relevant metrics, we will assume that Q is a query, D_{rel} the set of relevant documents for Q and D_{retr} the set of documents retrieved by our system.

3.6.1 Precision

Precision is a fundamental Information Retrieval metric evaluating how many retrieved documents have been true positives.

$$Precision(Q) = \frac{|D_{retr} \cap D_{rel}|}{|D_{retr}|}$$

3.6.2 Precision at K

Precision at K, e.g. P@25, accounts for the precision of the K highest ranked documents. As the end-goal of the project is to present a small set of highly relevant papers, we focused on this metric for relatively small K, such as K=5. Let us presume that $\{d_1, d_2, ..., d_{|D_{retr}|}\} = D_{retr}$ indicate the ranking of the retrieved documents.

$$\text{P@K}(Q) = \frac{|\{d_1, \ldots, d_K\} \cap D_{rel}|}{|\{d_1, \ldots, d_K\}|}$$

3.6.3 R-Precision

R-Precision is the precision at R where R is the number of relevant documents $|D_{rel}|$.

3.6.4 Mean Average Precision

Precision does not take ranked sequencing of the retrieved documents into account. The average precision is an attempt to do so.

$$\text{AvgPre}(Q) = \frac{\sum_{i=1}^{|D_{retr}|}(\text{P@i}(Q) \cdot \mathbb{I}[d_i \in D_{rel}])}{|D_{rel}|}$$

where $\mathbb{I}$ is the indicator function. The mean average precision, also called MAP, is the mean of average precisions over all queries. Thus we state that S is the set of queries.

$$\text{MAP}(S) = \frac{\sum_{Q \in S} \text{AvgPre}(Q)}{|S|}$$

3.7 QuickUMLS evaluation

The goal of this procedure is to inspect the potential of comparing documents and queries, referred to as ”topics”, based on the extraction of UMLS CUIs. In this context, a classification result is of the form: ”input x received CUI c” or ”input x did not receive CUI c”. Accordingly, we look into whether positive or negative documents are classified equally as the query – per CUI – and refer to this as ”topic-equal result”.

Leveraging QuickUMLS' approximate matching method, we analyzed the frequency of topic-equal results for positive documents against the frequency of topic-equal results for negative documents for each CUI c from a set of given CUIs. This is directly translated into Algorithm 1.

More on UMLS and QuickUMLS can be found in Subsection 4.6.1.

The set of positive documents P contains all documents marked to be relevant for the given topic query t. Note that the set of negative documents N is composed of randomly drawn documents explicitly marked as irrelevant, N_1, and randomly drawn unmarked documents, N_2, in equal proportions. As we observed an abundance of negative documents for each query, the size of N could be and was adapted as to level the size of P.

In Algorithm 1, said frequencies can be identified as the numerator and denominator of the *ratio* fraction, respectively. If the current CUI c is not attributed to the topic, i.e. the expression $(c\ in\ U_t)$ returns False, the desired quantities are the number of occurrences of c for positive/negative documents f_P/f_N subtracted from the number of positive/negative documents. Avoidance of division by 0 is ensured through a default denominator value of 1, illustrated by the max function inside of the conditional branches. The restriction of the method to a set of predefined CUIs, S, translates into the for-loop of the algorithm. The intermediary ratio of those quantities expresses how well positives can be distinguished from negatives for given topic t and current CUI c. Returning the average ratio $avgRatio$ over all ratios indicates the performance over all CUIs from S on topic t.

Input: Topic description t, set of modeling CUIs S
Output: Average ratio $avgRatio$
$U_t \leftarrow$ {CUIs for t }
$P \leftarrow$ {all articles marked relevant for t}
$N_1 \leftarrow$ {random articles irrelevant for t} with $|N_1| = \lceil |P|/2 \rceil$
$N_2 \leftarrow$ {random articles neither irrelevant nor relevant for t} with $|N_2| = \lfloor |P|/2 \rfloor$
$N \leftarrow N_1 \cup N_2$
$f_P \leftarrow$ map from CUIs to their #occurrences in P
$f_N \leftarrow$ map from CUIs to their #occurrences in N
$sumRatio \leftarrow 0$
for c ***in*** S **do**
  if c *in* U_t **then**
    $ratio \leftarrow \frac{f_P}{max\{f_N,1\}}$
  end
  else
    $ratio \leftarrow \frac{|P|-f_P}{max\{|P|-f_N,1\}}$
  end
  $sumRatio \leftarrow sumRatio + ratio$
end
$avgRatio \leftarrow \frac{sumRatio}{|S|}$
return $avgRatio$

Algorithm 1: Computes the average ratio of topic-equal positives and topic-equal negatives for set of CUIs and topic description

Chapter 4

Experiments

4.1 Data

4.1.1 MIMIC database

The MIMIC database is a deidentified collection of critical care patient data. It comprises not only a wide variety of over 2 million patient notes but also assignments of ICD-9 codes to over 45,000 patients. Note that ICD-9 codes are attributed to patients, not to notes and can represent diseases, symptoms, diagnoses, findings and circumstances. Depending on their motivation, the patient notes are classified into 15 categories such as "Discharge summary" or "Radiology". However, we did not distinguish based on their categorization. Over 650,000 attributions of ICD-9 to patients are included, involving roughly 7,000 distinct ICD-9 codes. The patient notes do not have a recurring structure, contain colloquial and technical abbreviations as well as orthographic mistakes and heavy use of special characters. The following is an example of a note from category "Nursing/other".

```
ccu nursing progress note(micu pt)
s: orally intubated and sedated
o: pls see carevue flowsheet for complete vs/data/
events
id: afeb. vanco dosing ^'d to q24hrs, cont on
piperacillin.
cv: hr 68-80s sr, freq pvcs, occ couplet. rare short
run of
?aivr. bp 100-150/70. pap 42-52/24-30. w 22. ci 3.9.
resp: weaned peep to 15, other vent settings unchanged.
sats 93-96%. sxn'd for sm amt white secretions.
gu: cont on lasix gtt at 10mg/jr. 1l neg. noted frank
hematuria this am, cleared within an hr. ua sent.
gi: tf at goal. no stool so rectal tube/mushroom cath
```

```
dc'd. was plugged. tf fs nepro w promod at 25cc/hr.
ms: appears comfortable, no agitation. no change
to fentanyl/versed gtts, has not req boluses.
social: family meeting planned for this afternoon.
a: stable
p: cont supportive care. family meeting. support to
family.
```

Binary classifiers have been trained and intrinsically evaluated on the patient notes and patient ICD-9 codes. Hence the determination of a note's label was of the kind "patient associated with this note has ICD-9 code k" or its relative negation. The intrinsic evaluation was performed on a random held out set of 15% of the notes.

4.1.2 TREC-CDS

The Text Retrieval Conference Clinical Decision Support track is a challenge aiming to retrieve relevant scientific articles based on patient note queries. We have been working on the dataset provided for TREC CDS 2015, containing a collection of over 700,000 documents and 30 queries, both in XML format. Each query, also called "topic", consists of a summary and a longer description. A sample query description looks the following:

```
A 31-year-old woman with no previous medical problems
comes to the emergency room with a history of 2 weeks
of joint pain and fatigue. Initially she had right
ankle swelling and difficulty standing up and walking,
all of which resolved after a few days. For the past
several days she has had pain, swelling and stiffness
in her knees, hips and right elbow. She also reports
intermittent fevers ranging from 38.2 to 39.4 degrees
Celsius and chest pain.
```

In addition, a selection of 1,000 documents per query is labeled as either definitely not relevant (0), potentially relevant (1) or definitely relevant (2).

4.2 ICD-9 code selection

The ICD-9 codes aiming to represent demonstrative concepts of patient notes, we needed to make sure they were carefully chosen with respect to quantity and quality.

Too many ICD-9 code predictions imply that some dimensions of the probability vector are likely to be not as expressive and dilute the cosine similarity

with noise. Too few ICD-9 code predictions lead to a lack of flexibility in describing patient states.

Furthermore, we needed to ensure that each selected ICD-9 code is associated *frequently enough* with patients as well as notes. At the same time, a too high frequency indicates that the labeling of an ICD-9 code is in shortage of information quantity or even is redundant.

Therefore we have looked at appearances of ICD-9 codes. Figure 4.1 indicates that most ICD-9 codes are attributed to less than 500 patients. On the other hand, we observed that there exists a group of ICD-9 codes occurring for more than a quarter and less than half of the patients. This group generally looked very tempting as it could be expressive yet not redundant.

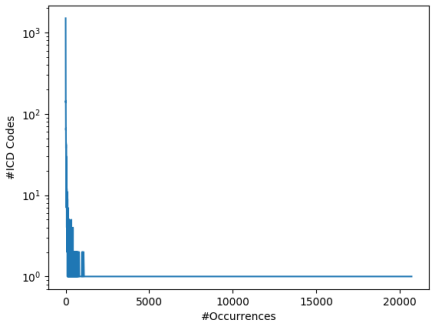

Figure 4.1: Occurrences of ICD-9 codes

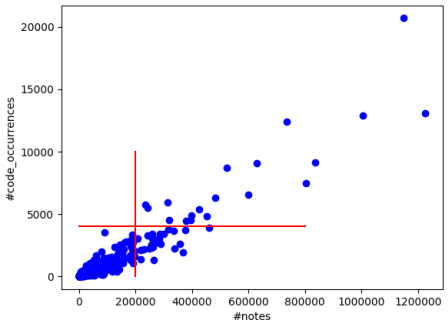

Figure 4.2: Thresholds for ICD-9 codes

When examining not only the number of patients associated with ICD-9 codes but also the number of associated notes, we obtained Figure 4.2. In order to satisfy the previously mentioned constraints, we opted for the heuristic of selecting the ICD-9 codes with a number of notes in between 10 and 60%, while being associated with 25 to 45% of the patients. 19 ICD-9 codes fulfilled the heuristic's conditions, after which we inspected their meaning, retractable in Table 4.1. We concluded that none of those was either "too general" nor "too specific" from an intuitive point of view.

4.3 Instrisic evaluation: classifier accuracy

Before vectorizing the input patient notes with term-frequencies, we applied some light preprocessing to the texts. In particular, we transformed them to be lower case and replaced special characters and punctuation by spaces. For each ICD-9 code, we selected all positive notes, i.e. notes from patients with given ICD-9 code in the training set. In addition, we used equally many negative notes, i.e. notes stemming from patients not associated with

Table 4.1: Explanations of selected ICD-9 codes

42731	Atrial fibrillation
2851	Acute posthemorrhagic anemia
2762	Acidosis
486	Pneumonia, organism unspecified
2859	Anemia, unspecified
2720	Pure hypercholesterolemia
53081	Esophageal reflux
V290	Observation for suspected infectious condition
5849	Acute kidney failure, unspecified
2449	Unspecified acquired hypothyroidism
25000	Diabetes mellitus without mention of complication, type II or unspecified type, not stated as uncontrolled
2724	Other and unspecified hyperlipidemia
4019	Unspecified essential hypertension
4280	Congestive heart failure, unspecified
496	Chronic obstructive pulmonary disease, unspecified
5990	Urinary tract infection, site not specified
41401	Coronary atherosclerosis of native coronary artery
51881	Acute respiratory failure
V053	Need for prophylactic vaccination and inoculation against viral hepatitis

that given ICD-9 code. The validation has been performed on a set of 50% positive and 50% negative notes as well. Given that a correct prediction has equal worth independently of its label prediction, we chose accuracy as the metric of evaluation. Accordingly, we have measured accuracy for each ICD-9 code of both a Naive Bayes and Logistic Regression classifier, as can be observed in figure 4.3. We used the open source scikit-learn library [1] for vectorization and classifier learning. Trading a much more computationally intensive model for only a slight gain in absolute performance [13] discouraged us from pursuing possible enhancement by a different model such as Support Vector Machines. In particular, even learning SVMs with linear kernels, a single ICD-9 code classifier could not be learnt on our setup within 48 hours, as compared to completion within less than 6 hours for Logistic Regression. The average accuracy for Naive Bayes averaged out at approximately 64.67% while Logistic Regression achieved 67.61%. Strikingly, Figure 4.3 yields two significant outliers with a precision of roughly 96% accuracy for Naive Bayes and 98% for Logistic Regression respectively. These are the ICD-9 codes V290 and V053. Upon examination of their description, one might suggest that their generality allows for application to almost any patients. Nevertheless, they did not occur as frequently. More precisely, they

[1] http://scikit-learn.org/stable/

have been associated with only 5,779 and 5,519 patients and 244,055 and 236,471 notes respectively. In addition, when looking into the probability distributions of the individual classifier on 1,000 random scientific articles from the TREC CDS dataset, most classifiers predicted in the shape of a Gaussian bell centered close to 0.5, such as 486 in Figure 4.4. However, V290 and V053 predictions tended far more towards 1, such as V290 in Figure 4.5. Both plots indicate Gaussians fitted to the occurrences of probability values. Said probability values are represented in the x-axis. This phenomenon is not comprehensible for us up to this point.

4.4 Cosine similarity inspection

In hope of having found a sufficiently expressive representation of the notes in the 19-dimensional probability vectors, we looked into the pairwise cosine similarities of 5,000 random articles from the TREC CDS dataset. In other

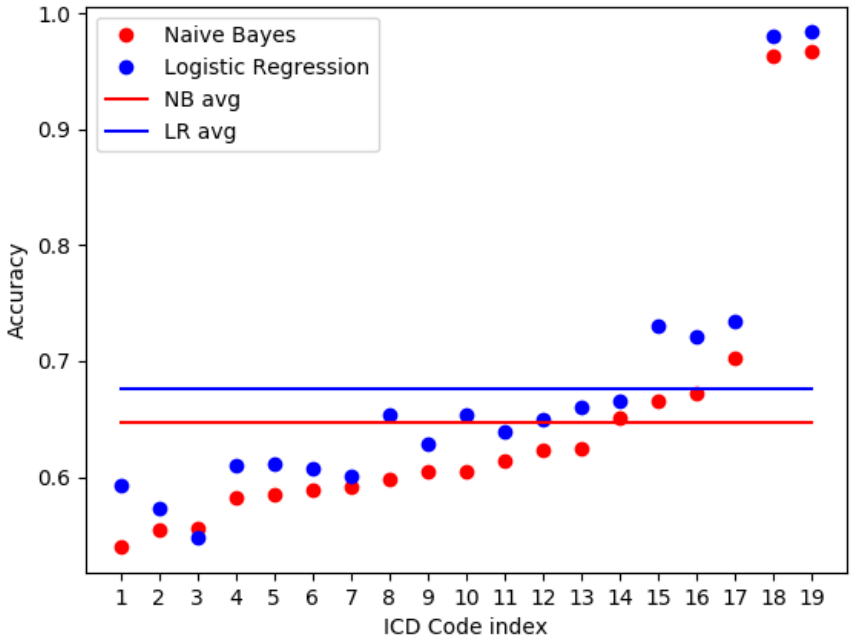

Figure 4.3: Classifier accuracy per ICD-9 code

words, each document d_i was represented as a 19-dimensional vector $x(d_i)$.

$$x(d_i) = \begin{bmatrix} \Pr(d_i \text{ has ICD-9 code 42731}) \\ \Pr(d_i \text{ has ICD-9 code 2851}) \\ \vdots \\ \Pr(d_i \text{ has ICD-9 code V053}) \end{bmatrix} \tag{4.1}$$

Subsequently, we computed the cosine similarity for each pair of such vectors. Figure 4.6 expresses how many similarities of pairs, in the x-axis, gave rise to a certain similarity value, in the y-axis. We concluded that the similarity was *surprisingly often surprisingly high*, as can be seen by the figure's long tail. This observation was disappointing, as it expresses that our classifiers might not "spread" the articles far enough apart. Nevertheless it is possible that similarities were small in general and would have to be amplified.

4.5 Extrinsic evaluation: TREC-Eval

The extrinsic evaluation was performed on the dataset provided along the TREC CDS 2015 challenge. Firstly, we determined a ranking of all articles per query. In order to do so, we merged each article's body, title and abstract XML contents and applied the same preprocessing as for the intrinsic evaluation. The topic descriptions of the queries underwent this preprocessing as well - we ignored the summaries of the queries as they did not provide additional information. An application of Apache Lucene's BM25 algorithm gave us a ranking of the 1,000 most relevant articles per query - this represented our baseline.

In order to instantiate our composite information retrieval, we computed the predictions and cosine similarities for the 1,000 pairs of documents already retrieved by BM25 and the respective query. This result was then used to

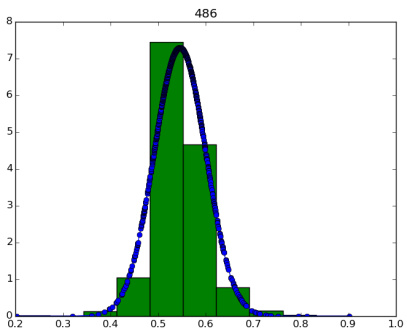

Figure 4.4: Probability distribution for 486

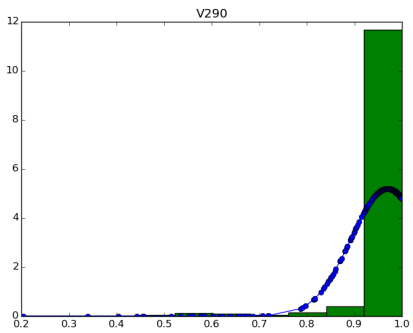

Figure 4.5: Probability distribution for V290

Table 4.2: TREC Eval results

λ	P@5	P@10	P@30
0	0.3467	0.3	0.2633
0.1	0.34	0.2933	0.26
0.2	0.3267	0.2933	0.2589
0.5	0.3	0.2833	0.22289
1	0.18	0.17	0.1767

compute our adapted ranking with varying λ. As they had performed consistently better in the intrinsic evaluation, we used the Logistic Regression classifiers to obtain probability vectors.

Based on a human-labeled set of relevance scores, the evaluation tool from TREC yields the results in Table 4.2. It is clearly observable that a greater λ, to be interpreted as higher emphasis on the classification similarity, leads to worse precision. In other the words, the baseline of $\lambda = 0$ outperforms our synthesized method, even in additional metrics such as MAP and R-Precision.

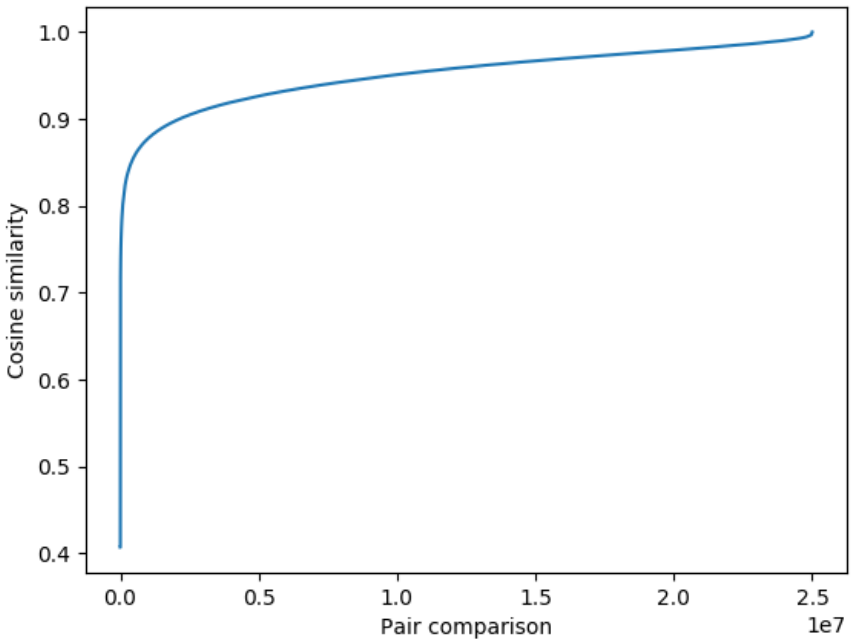

Figure 4.6: Cosine similarities of pairs of documents

The conclusion we drew from those results was that the classifier model lacks in power. In particular, the ICD-9 codes might not have been sufficiently distinguishing as concepts. Still, we did not gain insight in whether our general approach could be validated with a different explicit topic model.

4.6 Inspection of UMLS CUIs

4.6.1 Background

The Unified Medical Language System is a metathesaurus allowing for mapping and translations between many biomedical controlled vocabularies. In particular, UMLS provides Concept Unique Identifiers (CUIs), mapping codes to specific medical meanings, similar toICD-9 codes.

QuickUMLS is a tool for medical concept extraction. In other words, one of its key components is to return a set of relevant UMLS CUIs for a given text query. Similar tools, such as cTAKES[2] and MetaMap[3], predate QuickUMLS, yet it is significantly faster without considerable loss in either precision nor recall [11]. Generally, one text can obtain a specific UMLS CUI several times. We decided to simplify the experiment by binarily analyzing whether a text had obtained a CUI at least once.

4.6.2 Motivation

Relying on QuickUMLS was a cheap way of further exploring the potential of explicit topic modeling as our heuristic with ICD-9 code modeling had not flourished. We were looking to discover insights about the upper bound of topic modeling, the performance of our concepts translated to UMLS and the best possible set of concepts we could find. In order to evaluate performance, we investigated the average ratios from Algorithm 1 expressing the similarity of positive notes with the query compared to negative notes. We used the TREC CDS dataset from 2014 for this experiment.

4.6.3 Upper bound: adaptation of concepts to queries

Albeit not being a *theoretical* upper bound, we presumed that overfitting the CUIs to the topics individually should come close to it. By selecting the UMLS CUIs assigned to each topic as concept models for the evaluation of the respective topic we aimed to approach this bound. We chose the concept set S to be the set U_t of CUIs associated with the topic description with regard to Algorithm 1. Hence the else-branch never triggered. Executing the algorithm for each of the 30 topics we obtained an average ratio per topic

[2]http://ctakes.apache.org/
[3]https://metamap.nlm.nih.gov/

Table 4.3: Occurrences of UMLS CUIs among topic descriptions

#occurrences	1	2	3	4	5	6	7	8	12	21	28
#UMLS CUIs	721	96	44	22	11	17	3	1	1	1	1

of 4.5, while the minimum and maximum ratio per topic were 1.8 and 10.1. We inferred that a well-chosen set of concepts could still be indicative and expressive enough, as a 4-fold greater similarity of positives than negatives seemed substantial. We then continued by seeking an explicit set of concepts for the general case.

4.6.4 Set of concepts for all topics

Firstly, we looked into the potential of the concepts we had chosen for ICD-9 codes in UMLS CUIs. The 1-to-1 translation of ICD-9 codes to CUIs was facilitated by the metathesaurus provided by the National Cancer Institute[4]. Using default QuickUMLS retrieval settings, there was a single occurrence of any of 19 concepts over all of the 30 queries. No substantially different behaviour was observable under custom QuickUMLS settings. This led us to believe that the ICD-9 code translations were not suitable.

We decided that looking at mere occurrences of UMLS CUIs among topic descriptions should give us a reasonable indication of whether they can be expressive. The results from this investigation are recorded in Table 4.3.

Establishing the heuristic of each CUI appearing in at least 6 and at most 27 topic descriptions, we ended up with 23 CUIs. We proceed by naming this selection U_M.

We applied Algorithm 1 accordingly with $S \leftarrow U_M$ for 22 out of the 30 topics at random, for performance reasons. Figure 4.7 displays the algorithm's result for each of the $22 \cdot 23$ executions on the y-axis. In comparison to our "upper bound", the average ratio over all topics fell dramatically to 1.445 whereas minimum and maximum ratio dropped to 0.759 and 4.539. We observed that 44.86% of the ratios ended up within the range [0.8,1[. More importantly, investigation of the ratios per CUI per topic greater or equal to 1, portrayed as above the red threshold in the figure, returned a fraction of 0.336. We deduce that for only a third of CUI-topic pairs, the positive documents are at least as frequently topic-equal as the negatives documents. We distinguished two observations:

(1) The fraction is below 0.5.

(2) The fraction is not distinctly above 0.5.

[4]https://ncim.nci.nih.gov/ncimbrowser/

At first sight, (1) seems indicative for a mistake, as it performs worse than a random baseline. However, we suppose that this is a consequence of small sample size. In particular, the typical amount of documents marked as relevant was around 100. On the other hand, half of the negatives were drawn at random from a set of roughly 800,000 documents, hence substantial deviations of the ”average positive document” from the ”average document” are both possible and impactful. Moreover, the observation that many ratios lie in-between 0.8 and 1 enforce this hypothesis. We refer the reader again to Figure 4.7 for an illustration of this pattern. (2) implies the impropriety of this method for the purpose of explicit topic modeling, well before going into detail and measuring the actual precision via TREC-Eval. In particular, this notice translates into the fact that positive documents cannot be recognized substantially better as similar to the topic than negatives – which exactly describes our overarching goal.

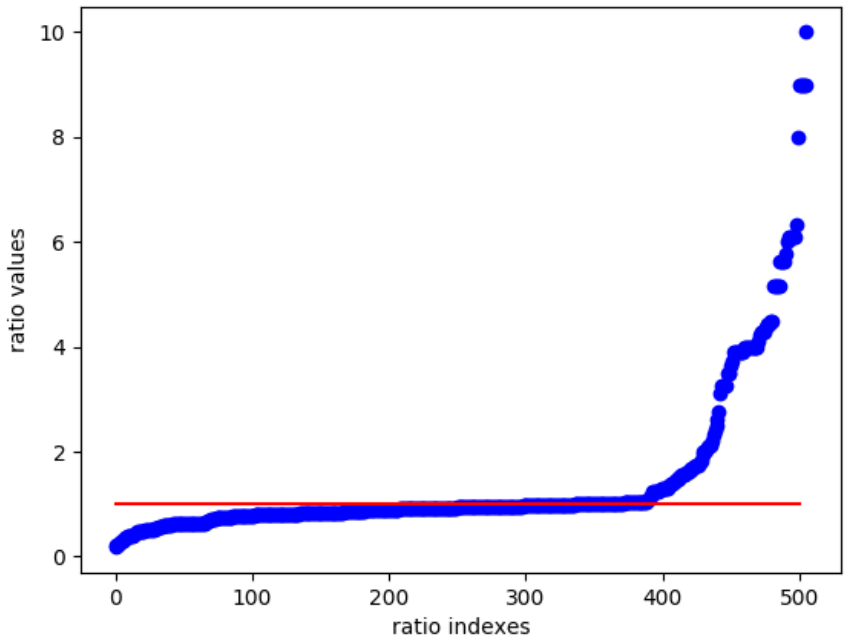

Figure 4.7: Evaluation ratios per topic per selected CUI

Chapter 5

Discussion

Summing up, we conclude that our ICD-9 code predictions from the MIMIC dataset are not suited for explicit topic modeling of the documents and queries of the TREC CDS dataset. We recognize four potential causes.

Firstly, the issue might lie in a substantial difference in the types of documents: diverse critical care patient reports in the MIMIC dataset for learning the classifiers, very short query descriptions and scientific articles from the TREC CDS dataset for evaluation of the classifiers. Inherent distinctions in length of documents as well as their content, most notably abbreviations, sentence structuring and terminology can be seen as potential barriers. Secondly, the restriction of concepts to ICD-9 codes might have been inappropriate. On the one hand, the fine granularity of more than 10,000 ICD-9 codes might be disadvantageous as compared to groupings of ICD-9 codes. Using sets of identifiers instead of sole identifiers was an option we did not pursue. On the other hand, the ICD-9 Codes attributed in the MIMIC database might be inherently biased towards specific requirements, e.g. indications by insurances. Thirdly, our heuristic with respect to determining our selection within ICD-9 codes might have been flawed. In particular, the insignificant variance in cosine similarities suggests that they might not be distinctive enough. Lastly, the classifier accuracy might have been too low. Enhancements such as a more elaborate predictive model or natural language processing techniques like stemming or lemmatization could have led to improvements on accuracy and finally in power of retrieval. We rejected this idea due to the low rise in accuracy upon transitioning from Naive Bayes to Logistic Regression.

At the same time we are confident in saying that attributing UMLS CUIs to the queries and scientific articles via QuickUMLS is not advisable method for explicit topic modeling and similarity measurements in the TREC dataset. Occurrences of single CUIs in the queries are infrequent in the topic descriptions of relatively short size, leading to a lack of indicative potential of single

CUIs. Looking into sets of CUIs would certainly be of interest for future investigations.

We are left with the uncertainty whether single concept identifiers can be useful for the CDS challenge as topic models. If they are not and concept identifiers need to be regrouped in order to be useful - the explicit topic modeling might lose its appeal. Such groupings of identifiers can easily lead to hardly interpretable sets. Precisely the initial advantage of explicit topic modeling over latent topic modeling would be dwindling in that case.

Chapter 6

Conclusion

Our experiments have shown that a baseline of BM25 overperforms a combination of BM25 and the cosine similarity of ICD-9 code predictions in the TREC CDS challenge. The topic models have been hand-picked according to heuristics considering occurrences of ICD-9 codes. The results indicates that the classifiers, trained on the MIMIC dataset, lack in expressive power with respect to the TREC documents and queries. At the same time, vectorizing texts by their relatedness to a set of UMLS CUIs has shown to be ineffective as well.

On the one hand, latent topic modeling has demonstrated to perform well in the TREC CDS challenge. On the other, explicit topic modeling's interpretative transparency and understandability remains appaeling. We believe that explicit topic modeling is most useful when topics are single or few concept identifiers and it is unclear whether such concepts can be helpful with regard to the challenge. Our empirical analysis suggests the answer is no. The outlook of investigating this question furtherly by replacing our heuristics with respect to the choice of concepts with more extensive experiments clearly seems interesting.

Bibliography

[1] Saeid Balaneshinkordan, Alexander Kotov, and Railan Xisto. Wsu-ir at trec 2015 clinical decision support track: Joint weighting of explicit and latent medical query concepts from diverse sources. In *TREC*, 2015.

[2] Shariq Bashir and Andreas Rauber. Improving retrievability of patents in prior-art search. In *European Conference on Information Retrieval*, pages 457–470. Springer, 2010.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[4] George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971, 1987.

[5] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJcAI*, volume 7, pages 1606–1611, 2007.

[6] Simon Greuter, Philip Junker, Lorenz Kuhn, Felix Mance, Virgile Mermet, Angela Rellstab, and Carsten Eickhoff. Eth zurich at trec clinical decision support 2016.

[7] Joshua Aaron Hansen. Probabilistic explicit topic modeling. 2013.

[8] Sadid A Hasan, Siyuan Zhao, Vivek Datla, Joey Liu, Kathy Lee, Ashequl Qadir, Aaditya Prakash, and Oladimeji Farri. Clinical question answering using key-value memory networks and knowledge graph. TREC, 2016.

[9] Peder Olesen Larsen and Markus Von Ins. The rate of growth in scientific publication and the decline in coverage provided by science citation index. *Scientometrics*, 84(3):575–603, 2010.

[10] Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC–3. In *Overview of the Third Text REtrieval Conference (TREC–3)*, page 109–126. Gaithersburg, MD: NIST, January 1995.

[11] Luca Soldaini and Nazli Goharian. Quickumls: a fast, unsupervised approach for medical concept extraction. In *MedIR Workshop, SIGIR*, 2016.

[12] Yang Song, Yun He, Qinmin Hu, and Liang He. Ecnu at 2015 cds track: Two re-ranking methods in medical information retrieval. In *Proceedings of the 2015 Text Retrieval Conference*, 2015.

[13] Thierry Verplancke, Stijn Van Looy, Dominique Benoit, Stijn Vansteelandt, Pieter Depuydt, Filip De Turck, and Johan Decruyenaere. Support vector machine versus logistic regression modeling for prediction of hospital mortality in critically ill patients with haematological malignancies. *BMC medical informatics and decision making*, 8(1):56, 2008.

[14] Le Zhao and Jamie Callan. Term necessity prediction. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 259–268. ACM, 2010.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Structured Information Retrieval of Natural Language Supporting Clinical Decision-making

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Klein	Kevin

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date

Luxembourg, March 7, 2017

Signature(s)

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.