Incorporating Semantic Knowledge with MRF Term Dependency Model in Medical Document Retrieval

Zhongda Xie(✉), Yunqing Xia, and Qiang Zhou

Department of Computer Science, TNList,
Tsinghua University, Beijing 100084, China
{xzd13,yqxia,zq-lxd}@tsinghua.edu.cn

Abstract. Term dependency models are generally better than bag-of-word models, because complete concepts are often represented by multiple terms. However, without semantic knowledge, such models may introduce many false dependencies among terms, especially when the document collection is small and homogeneous(e.g. newswire documents, medical documents). The main contribution of this work is to incorporate semantic knowledge with term dependency models, so that more accurate dependency relations will be assigned to terms in the query. In this paper, experiments will be made on CLEF2013 eHealth Lab medical information retrieval data set, and the baseline term dependency model will be the popular MRF(Markov Random Field) model [1], which proves to be better than traditional independent models in general domain search. Experiment results show that, in medical document retrieval, full dependency MRF model is worse than independent model, it can be significantly improved by incorporating semantic knowledge.

Keywords: Semantic knowledge · Term dependency model · Markov random field · Medical document retrieval

1 Introduction

Nowadays, term dependency retrieval models [1–3] generally outperform the traditional bag-of-word models. This is because a complete concept in a query is often represented by multiple terms; so, the occurrence of multiple dependent terms in a document can provide a stronger evidence that the document is relevant with the query. In general domain search, these models are significantly better. MRF model [1] proposed by Metzler and Croft is among the best, which indicates that full dependency model is better than sequential dependency and full independency models. However, in some specific areas, like medical document retrieval, the data collection is homogeneous, the documents are filled with formalized terminologies, and these term dependency models will introduce too many false dependencies.

Take the following two queries as examples:

J. Li et al. (Eds.): NLPCC 2015, LNAI 9362, pp. 219–228, 2015.
DOI: 10.1007/978-3-319-25207-0_19

(Query1): coronary artery disease and myocardial infarction.
(Query2): abdominal pain and helicobacter pylori and cancer.

In Query1, *coronary artery disease* is a kind of disease, while *myocardial infarction* is another, and the patient wants to know how to distinguish the two. The main difference between medical queries and general domain queries is that, the concepts in medical domain are represented by formalized terminologies. That is to say, if a medical concept is to appear in a document, the words are most likely to be in the same order as the query, and tend to be close to each other, with no other word inserted inside. So, the good dependency detected for this query is *coronary artery disease* and *myocardial infarction*; while traditional MRF model will generate other dependency relations like *coronary infarction*, *myocadial disease*, *artery myocardial* and so on, which are all false dependency relations. The main reason of such difference is that, in general domain queries, most single word is meaningful enough, so arbitrary combination of the words is likely to represent a complete concept, which can be related to the query; But in medical domain, the multiple words belonging to a terminology must get together to represent the right concept, while a single word can hardly do.

In Query2, *helicobacter pylori* is a kind of bacteria, and it CAUSES *cancer*, now the patient wants to know whether *abdominal pain* is the SYMPTOM of such *cancer*. Now that we know there exist a CAUSES relation between *helicobacter pylori* and *cancer*, and a SYMPTOM_OF relation between *abdominal pain* and *cancer*, we can assign accurate dependencies to these concept pairs. And we will leave *abdominal pain* and *helicobacter pylori* independent, for they don't have direct relations to each other. Such kind of detection reply highly on semantic knowledge web, this is why we will incorporate it with state-of-art term dependency models.

In our work, we use UMLS(Unified Medical Language System) as our knowledge base, and SemMedDB(built on the same concepts in UMLS) as our semantic web. We first use MetaMap program to extract medical concepts in the query, and add the concept as matching unit. We make the comparison between two ways of matching the concept: using the exact sequence of the concept words; and detecting unordered concept words in a window of document text, as the traditional MRF model will do. Then, we use the SemMedDB database to detect related concept pairs, and take such pairs as additional matching units(e.g. [*helicobacter pylori*, *cancer*]). Finally, we extract the words indicating relationship between concept pairs from the SemMedDB, and put the related concept pairs together with their relation word as matching units(e.g. [*helicobacter pylori*, *CAUSES*, *cancer*]). The experiments are conducted on CLEF eHealth 2013 data set, results show that our model is better than BM25 and the three variants of traditional MRF model(full independency, sequential dependency, full dependency).

The rest of this paper is organized as follows. In section 2 we will summarize the related works. The details of our methods are in section 3, while the experimental results and analysis will be in section 4. We make the conclusion in section 5.

2 Related Work

Term dependency models generally have advantage over bag-of-word models, because the co-occurrence of dependent words in a document has a stronger sign that the document is relevant. So, much work has been done in this direction.

Xu et al. [2] introduced a extend BM25 model. It extends the traditional BM25 equation from single word based to n-gram based, and sums up the scores of the n-grams, ranging from all lengths and all start positions. This model is very simple, but will always outperform the basic IR systems. However, this model cannot accurately catch the dependency among separated words, due to its n-gram based nature. Take the Query2 from section 1 as an example, it cannot precisely get the unit *abdominal pain cancer*, which has a SYMPTOM_OF relation between the two concepts, and it has to get the whole query as a 7-gram in order to capture this relation.

Later, Many works [3,4] rely on text parse trees to extract more accurate dependencies. However, these models still only consider about pairs of words. Park et al. [4] uses a language model to make use of the parse tree, in which a document generates a parsing path in a certain probability, and that path will generate the query. They treat all the paths equally important, which will introduce many noisy word dependencies, and the parsing process on the whole document is rather computation intensive.

Metzler and Croft [1] then propose a MRF model to represent the term dependencies. MRF model is very flexible, by defining different kinds of feature functions, they can put various features(single word; neighboring, ordered dependent words; separate, unordered words and so on) together in one universal model. This model has three basic variants: full independency model, sequential dependency model, full dependency model. They differ on what combination of words is dependent. The model works out very well in general domain search, and the three variants are in ascending order of precision as listed previously. However, the full dependency model is bad in small, homogeneous collections(like medical report collection), as has been explained in section 1. And it needs appropriate knowledge to extract precise dependency relations.

In medical document retrieval, much work has shown that concept detection and concept pair relation can be helpful. Qi et al. [5] uses a concept-based model, and experiment with many popular methods for word based models, and find out that vector space model together with pseudo relevance-feedback works well. Later, researchers go further to concept level relations. Khoo et al. [6] make an attempt to use the cause-effect relation between concepts, which they think is the most helpful relation. Lee et al. [7] use ontology relations between concepts, like *IS_A* and *CO-OCCURS_WITH*. However, Vintar et al. [8] find that those are coarse-grained relations, which have no real meaning, so they want to use fine-grained relations extracted from semantic web, like *TREAT* and *SYMPTOM_OF*. They use such relations as a document filter in a boolean manner, which only improves traditional models slightly. So, Xia et al. [9] introduce a way to represent the query and documents in relation level, and compute their

relevance in both word level and relation level, the final score is a combination of the scores in two levels.

Inspired by above promising works, we want to incorporate medical semantic web and its corresponding techniques with MRF model, which is flexible and rather powerful in general domain. We want to assign accurate dependencies to the words, thus improving precision of the system.

3 Methodology

In order to incorporate the medical semantic knowledge with MRF model. First we need to detect the medical concepts and the find-grained relations among them. After that, a core problem is how we will reflect such knowledge in the model, and the flexible feature functions of MRF provides a way. In the following sections, we will discuss about the details of our modules.

3.1 Concept Detection

The knowledge base we use is UMLS(Unified Medical Language System), developed by NLM(National Library of Medicine) of U.S.. It integrates medical concepts from different authoritative resources, merges the medical terminologies with the same meaning into one concept, and gives each concept a unified, unique identifier.

In order to extract the concepts, we use a program called SemRep, which is developed based on the UMLS ontology. It will split the text into separated parts, and map each part into a concept(actually a unique concept identifier, called CUI) in UMLS. For example, in Query2 of section 1, we will get the result as following(we have changed the original output format for the sake of easy understanding):

[abdominal pain: C0000737] and [helicobacter pylori: C0079488] and [cancer: C0004382]

The brackets indicate split parts of the text, and the sequence after the terminology is the CUI, the unique identifier of the concept.

The output of the program may have several ways of splitting the text and various mappings from a terminology to the concept in the UMLS database. We only choose the best parsing result(indicated by the program in terms of accurate probability), in order to make our detection process precise enough.

3.2 Concept Relation Extraction

The concept relation in our work does'n mean coarse-grained relations, which includes the common *IS_a* and *CO-OCCURRES_WITH* relations. They only indicate the ontology hierarchical structures, but not the real semantic relations. What we need is the find-grained relations like *TREATS* and *DIAGNOSES*, which represent meaningful relations in medical domain.

The semantic web we use is SemMedDB, it is built on top of UMLS ontology, so the medical concept extracted by SemRep can find its match in SemMedDB, using the CUI assigned to it. SemMedDB integrates around 70 million labeled medical domain sentences as its resource. For each sentence, there will be several pairs of related concepts, each called a *predication*. For each predication, we can follow the information in it to extract its corresponding two concepts, as well as the relation type between them. Take the following sentence as an example:

It is said that the helicobacter pylori often causes cancer, so, is my recent abdominal pain a possible symptom of my cancer?

From SemMedDB, we can know that there is a predication [*helicobacter pylori, cancer*], and the relation type is *CAUSE*; we can also find another predication [*abdominal pain, cancer*], and the relation type is *SYMPTOM_OF*.

So, we have shown how to get the related concept pairs in a sentence in the database. But we are faced with is different: we have a query, we extract all its concepts in the way described in section 3.1, enumerate all the possible pairs, and we want to find out whether each concept pair is related(or belong to a predication). And the solution goes in the opposite direction against the way we get predictions from a labeled sentence in database. We get the CUIs of the two concepts and find their entries in SemMedDB, then search the predication table to find whether they appear in the same predication; we can use the predication ID to further get the labeled sentence that the predication comes from. In this way, we can start from two concepts in the query, and find all the possible relations that have been labeled in the SemMedDB database.

There is other information that we can extract. For a related concept pair, we can extend it into a triple, containing the keyword that indicates the relation between them. For instance, for the pair [*helicobacter pylori, cancer*], it can be extended to [*helicobacter pylori, causes, cancer*], [*helicobacter pylori, caused, cancer*], [*helicobacter pylori, lead to, cancer*] and so on, all these extensions reflect the *CAUSE* relation between the concepts. We will use these triples as matching unit in our last module, the occurrence of the keyword *causes*, *caused* and *lead to* in the document can give a stronger evidence that it is talking about the two concepts, specifically with the wanted relation *CAUSE*.

In order to obtain the keywords indicating a certain relation, we need to preprocess the SemMedDB database. We traverse the SENTENCE_PREDICTION table, which contains all the predications. For each predication, we follow the link in it to extract the relation type corresponding with it; we also look for the sentence that produce the predication, use the position information from SENTENCE_PREDICATION table to find the exact keyword that indicates the relation.

Finally, we obtain fifty-seven relation types from the database, and remain forty-eight of them(the fine-grained relations only), with half positive and half negative. The twenty-four positive relations are: *ADMINISTERED_TO, AFFECTS, ASSOCIATED_WITH, AUGMENTS, CAUSES, COEXISTS_WITH, COMPLICATES, CONVERT_TO, DIAGNOSES, DIS-*

RUPTS, INHIBITS, INTERACTS_WITH, LOCATION_OF, MANIFESTATION_OF, METHOD_OF, OCCURS_IN, PART_OF, PRECEDES, PROCESS_OF, PRODUCES, PREDESPOSES, STIMULATES, TREATS, USES, PREVENTS. And the negative relations are ones that begin with *NEG_*, for example, the negative relation against *CAUSES* is *NEG_CAUSES*, which means something is not the cause of a disease.

The negative category of relations is rather useful. In case a patient wants to find out whether a kind of bacteria is the cause of a illness, while actually it isn't. If we only have the positive relations, the bacteria and the illness will obviously not appear in the *CAUSE* keyword list, and we will ignore the patient's intent. Only by remaining the *NEG_CAUSE* relation, can we understand what the patient wants.

3.3 Feature Function

In term dependency models, the three main characters will make the difference:

1. The number of words together as a matching unit.
2. Whether the words in the document should appear in the same order as the query or not.
3. The text window in the document that we detect related multiple words.

We will talk about the choice of traditional MRF model briefly, and explain our strategy when concept and relations have been detected.

In MRF model, there are three types of combination of query words: single word, sequential words, separated words. Different type leads to different strategies, and it is reflected by using different feature functions. Feature function is a very flexible character of the MRF model, we can merge the above various types in a universal way, and finally sum up the score of all the functions. The work in [1] uses Indri as search engine, which provides convenient syntax to represent all the feature functions.

The table below is the feature functions used in [1] and their corresponding Indri search queries. In the column of *Indri Query*, the #1() means the words inside needs to appear in order and consecutively, the $\#uwN()$ means the words inside only need to appear in a text window of size N, with no strict order required.

Table 1. Feature functions and associated Indri queries used in traditional MRF model

Type	Feature	Indri Query
Single Word	$f_T(q_i, D)$	q_i
Ordered Phrase	$f_O(q_i, q_{i+1}, ..., q_{i+k}, D)$	$\#1(q_i, q_{i+1}, ..., q_{i+k})$
Unordered Phrase	$f_U(q_i, ..., q_j, D)$	$\#uwN(q_i, ..., q_j)$

In tradition MRF model, q_i indicates the ith word in the query. Consecutive sequential words will use two feature functions, the ordered phrase type(which appear ordered and consecutively in the document), and the unordered phrase type(which only needs to appear in a text window). On the other hand, Separate words in the query, like (q_1, q_2, q_4), can only be matched through the unordered phrase function.

In our model, we have different strategies for single concept and concept pairs, which will be explained separately in the following sections.

(1)Single Concept Feature
The number of words is not fixed, because the basic unit is concept, so the number will be up to the length of the current concept. The concept words appear consecutively in the query. If it is in the traditional MRF model, two features will be computed, both as ordered and unordered phrase. But for a medical concept, we only compute its score in a ordered way. This is mainly due to the formalization feature of the medical terminologies, which has been explained in Section 1. The feature functions are listed in table 2, and we only choose *Ordered Phrase* feature.

Table 2. Feature functions and Indri queries used in our model

Type	Feature	Indri Query
Single Word	$f_T(q_i, D)$	q_i
Ordered Phrase	$f_O(c_1, c_2, ..., c_{\|c\|}, D)$	$\#1(c_1, c_2, ..., c_{\|c\|})$
Combined Phrase	$f_C(p_1, p_2, D)$	$\#uwN(\#1(p_1)\#1(p_2))$

p_i represents the ith phrase, which is also $c_{1,1}, c_{1,2}, ..., c_{1,|c_1|}$.

(2)Concept Pair Feature
Concept pair is made up of two concepts. In order to keep up with the formalization of medical terminologies, the single concept alone is still computed in an ordered way. However, in concept level, the two concepts can be unordered. For example, the pair *[some_disease, some_illness]* may appear in the document as *some_disease causes my some_illness*, or *my some_illness is caused by some_disease*. So, we define a *Combined Phrase* type for such kind of feature.

Later, we extract keywords for each relation type, and this is easily realized by adding the keyword in the *Combined Phrase* type.

4 Experiments

We use CLEF 2013 eHealth Lab Medical Retrieval data set as our collection. 50 queries are provided officially as test set, and they cover a wide range of health topics.

The evaluation metrics used are the popular ones in information retrieval: (1) p@10: precision considering only the top 10 returned documents. (2)nDCG@10: normalized Discounted Cumulative Gain, also assessed at top 10 documents returned.

4.1 Experiment 1: Feature Function for Single Medical Concept

In algorithms described in section 3, we only use ordered phrase feature for single medical concept. Actually, before we go on to the next steps, we make the comparison between ordered phrase feature and unordered phrase feature for single concept. And the experimental result is as follows:

Table 3. Results using different features for single medical concept

Features	P@10	nDCG@10
Ordered Phrase	0.4960	0.5043
Unordered Phrase	0.4840	0.4967
Ordered and Unordered Phrase	0.4880	0.4963

Result shows that only using ordered phrase feature is the best, this looks up to our analysis of the formalized feature of medical concepts. In fact, we find some examples indicating unordered phrase feature is not fit for medical concepts. In the returned documents, the model matches *coronary heart disease and coronary artery revascularization* as relevant to query *coronary artery disease*. It shows matching a medical concept in strict order is very important.

4.2 Experiment 2: Results of Different Models

We compare our revised models against the three variants of MRF model. The abbreviation of each model is listed below:

-**MRF(I)**: full independency model of traditional MRF.
-**MRF(S)**: sequential dependency model of traditional MRF.
-**MRF(F)**: full dependency model of traditional MRF.
-**MRF(C)**: our model using features for single medical concept(only ordered phrase feature).
-**MRF(CR)**: our model using both features for single medical concept and concept pairs.
-**MRF(CR-EX)**: extended model(using relation keyword) of **MRF(CR)**.

The results is as table 4.

The first three models are the variants of the traditional MRF model, they introduce more and more word dependencies. The full dependency model(**MRF(S)**) works out best in general domain, but worst among the three in

Table 4. Results using different models

Model	P@10	nDCG@10
MRF(I)	0.4940	0.5087
MRF(S)	0.4780	0.5028
MRF(F)	0.4580	0.4762
MRF(C)	0.4960	0.5100
MRF(CR)	0.5060	0.5193
MRF(CR-EX)	0.5100	0.5203

medical domain. Without careful assigned dependencies, term dependency model cannot work well in small, homogenous collections like medical report collection. **MRF(C)** introduces medical semantic knowledge, only uses the extracted medical concepts, and only in a ordered phrase form. The accurately detected dependency relation significantly regains the precision of the system, especially against **MRF(F)** model. Then, **MRF(CR)** model uses related concept pair as a matching unit, which can give a stronger evidence that the document is talking about both of them. The extended model also improves a little, better ways of using the extended keyword should be proposed, and the gain will be larger.

5 Conclusion

In this work, we find out that, the main reason for the failure of traditional MRF term dependency model in medical domain, is the formalized feature of the medical terminologies. Thus, we make use of medical semantic knowledge to extract medical concepts, and only remain dependency relations of these concepts in a ordered form. The accurate dependencies in our model provides significant gain in precision of our system against the full dependency MRF model(**MRF(F)**). We also process the semantic web database to detect relations between concept pairs, and define a combined way of using ordered concept phrase and unordered concept pair. The related concept pair can give stronger evidence that the document is about both the concepts and the relation between them, thus again improves our system.

The extended model is not so good, one reason is that the keyword list need to be cleaned. Currently, the multiple relations existing between a concept pair are treated as equally important, this can be another reason that the extended model doesn't perform as we want. So, in the future, we will develop algorithms to assign different weights to the relations.

Acknowledgement. This work is supported by National Science Foundation of China(NSFC: 61272233). We thank the anonymous reviewers for the valuable comments.

References

1. Metzler, D., Croft, W.B.: A markov random field model for term dependencies. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Infromation Retrieval, SIGIR 2005, pp. 472–479. ACM, New York (2005)
2. Xu, Jun, Li, Hang, Zhong, Chaoliang: Relevance ranking using kernels. In: Cheng, Pu-Jen, Kan, Min-Yen, Lam, Wai, Nakov, Preslav (eds.) AIRS 2010. LNCS, vol. 6458, pp. 1–12. Springer, Heidelberg (2010)
3. Gao, J., Nie, J.-Y., Wu, G., Cao, G.: Dependency language model for information retrieval. In: Proc. 27th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 170–177 (2004)
4. Park, J.H., Croft, W.B., Smith, D.A.: A quasi-synchronous dependency model for information retrieval. In: Proceedings of the 20th ACM International Conference on Information and Knowledge management. CIKM 2011, pp. 17–26. NY, USA, New York (2011)
5. Qi, Y., Laquerre, P.F.: Retrieving medical records: NEC Labs America at TREC 2012 medical track. In: TREC 2012, Gaithersburg, Maryland, NIST(2012)
6. Khoo, C.S.G., Myaeng, S.H., Oddy, R.N.: Using cause-effect relations in text to improve information retrieval precision. Inf. process. Manage. **37**(1), 119–145 (2001)
7. Lee, J., Min, J.K., Oh, A., Chung, C.W.: Effective ranking and search techniques for web resources considering semantic relationships. Inf. Process. Manage. **50**(1), 132–155 (2014)
8. Vintar, S., Buitelaar, P., Volk, M.: Semantic relations in concept-based cross-language medical information retrieval. In: Proceedings of ECML/PKDD workshop on Adaptive Text Extraction and Mining. ATEM (2003)
9. Xia, Y., Xie, Z., Zhang Q., et al.: Cannabis_TREATS_cancer: Incorporating Fine-Grained Ontological Relations in Medical Document Ranking. Communications in Computer & Information Science (2014)