Triple based Background Knowledge Ranking for Document Enrichment

Muyu Zhang, Bing Qin*, Ting Liu, Mao Zheng
Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
{myzhang,qinb,tliu,mzheng}@ir.hit.edu.cn

Abstract

Document enrichment is the task of retrieving additional knowledge from external resource over what is available through source document. This task is essential because of the phenomenon that text is generally replete with gaps and ellipses since authors assume a certain amount of background knowledge. The recovery of these gaps is intuitively useful for better understanding of document. Conventional document enrichment techniques usually rely on *Wikipedia* which has great coverage but less accuracy, or *Ontology* which has great accuracy but less coverage. In this study, we propose a document enrichment framework which automatically extracts "$argument_1, predicate, argument_2$" triple from any text corpus as background knowledge, so that to ensure the compatibility with any resource (e.g. *news text*, *ontology*, and *on-line encyclopedia*) and improve the enriching accuracy. We first incorporate source document and background knowledge together into a triple based document-level graph and then propose a global iterative ranking model to propagate relevance score and select the most relevant knowledge triple. We evaluate our model as a ranking problem and compute the MAP and $P\&N$ score to validate the ranking result. Our final result, a MAP score of 0.676 and $P\&20$ score of 0.417 outperform a strong baseline based on search engine by 0.182 in MAP and 0.04 in $P\&20$.

1 Introduction

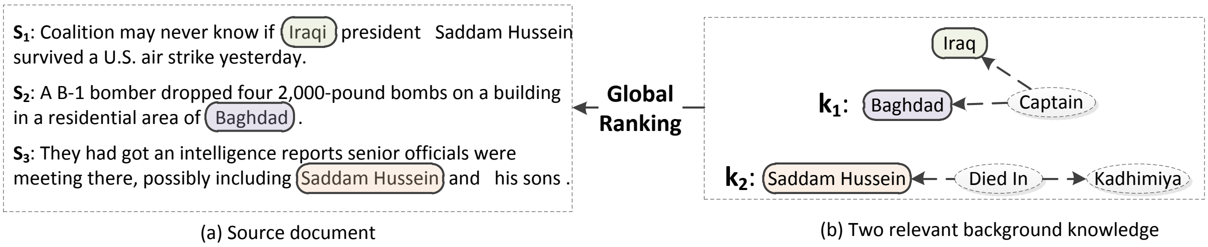
Document enrichment is the task to acquire background knowledge from external resources and recover the omitted information automatically for certain document. This task is essential because authors usually omit basic but well-known information to make the document more concise. For example, author omits "*Baghdad is the captain of Iraqi*" in the text of Figure 1 (a), which is well-known to readers. During reading process, these gaps will be automatically plugged effortlessly by the background knowledge in human brain. However, the situation is different for machine because it lacks the ability to acquire and select the proper background knowledge, which limits the performances of certain NLP applications. Document enrichment has been proved helpful in these tasks such as web search (Pantel and Fuxman, 2011), coreference resolution (Bryl et al., 2010), document cluster (Hu et al., 2009) and entity disambiguation (Bunescu and Pasca, 2006; Sen, 2012).

In the past, there are mainly two kinds of document enrichment researches according to the resource they relying on. The first line of works make use of *WikiPedia*, the largest available on-line encyclopedia as resource and link the entity (e.g. *Baghdad*) of document to its corresponding Wiki page (e.g. *Baghdad* [1] in WikiPedia), so that to enrich the document with the context of Wiki page (Bunescu and Pasca, 2006; Cucerzan, 2007; Han et al., 2011; Kataria et al., 2011; Sen, 2012; He et al., 2013). Despite the great success of these methods, there remain a great challenge that not all information in the linked Wiki page is helpful to the understanding of corresponding document. For example, the Wiki page of *Baghdad* contains lots of information about city history and culture, which are not quite relevant to the semantic of context in Figure 1 (a). So treating the whole Wiki page as the enrichment to document may cause noise

*Corresponding author.

[1] http://en.wikipedia.org/wiki/Baghdad

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 917–927, Dublin, Ireland, August 23-29 2014.


S1: Coalition may never know if Iraqi president Saddam Hussein survived a U.S. air strike yesterday.
S2: A B-1 bomber dropped four 2,000-pound bombs on a building in a residential area of Baghdad.
S3: They had got an intelligence reports senior officials were meeting there, possibly including Saddam Hussein and his sons.
Global Ranking
Iraq
k1: Baghdad
Captain
k2: Saddam Hussein
Died In
Kadhimiya

(a) Source document

(b) Two relevant background knowledge

Figure 1: An example of document enrichment with background knowledge: (a) source document talking about a U.S. air strike aiming at Saddam in Baghdad (b) two important relevant information, which is omitted in source document but acquired by our model and enriched as background knowledge .

problem. Another line of works rely on the Ontologies constructed with supervision or even manually which have great accuracy but less coverage (Motta et al., 2000; Passant, 2007; Fodeh et al., 2011; Kumar and Salim, 2012). Besides, these methods usually rely on special ontology which is rather difficult to construct and in turn limits the coverage and application of these methods.

Ideally, we would wish to integrate both coverage and accuracy, where an triple based background knowledge ranking model may help. Our framework extracts knowledge from any corpus resource including WikiPedia to ensure coverage and present knowledge as "$argument_1, predicate, argument_2$" triple to reduce noise. This model ranks background knowledge triples according to their relevance to the source document. The key idea behind the model is that document is constructed by several units of information, which can be extracted automatically. For every background knowledge b extracted automatically from a relevant corpus, the more units are relevant to b and the more important they are, the more relevant b becomes to the source document. Thus, we extract both source document information and background knowledge automatically and present them together in a document-level graph. Then we propagate the relevance score from the source document information to the background knowledge during an iterative process. After convergence, we obtain the $Top\ n$ relevant background knowledge, rather than retrieving all of them without filtering.

To evaluate our model, we use ACE [2] corpus as source documents and output the ranked list of background knowledge. Then we train three annotators to check the ranking result and annotating whether certain knowledge is relevant to corresponding source document separately. We totally annotated more than 7000 background knowledge by three annotators. We evaluate their annotation consistence by computing the $Fleiss'\ Kappa$ (Fleiss, 1971), a famous criterion in multi-annotator consistence evaluation. We achieve a $Fleiss'\ Kappa$ of value 0.8066 in best situation and 0.7076 in average, which indicates the great consistence between three annotators. The ranking result is evaluated with MAP score and $P\&N$ score (Voorhees et al., 2005). We finally achieve a MAP score of 0.676 and $P\&20$ score of 0.417 in Top 20 background knowledge, which are higher by 0.182 and 0.04 than a strong baseline based on search engine. We also evaluate the effect of the automatically extraction to source document and background knowledge, which is key to the performance of our method in real application.

2 Triple Graph based Document Representation

We believe that different parts of document are related to each other, rather than isolated. Hence, we propose a *triple graph* based document representation to incorporate source document information and background knowledge. In this presentation, "$argument_1, predicate, argument_2$" triple serves as node and the edge between nodes indicates their semantic relevance. In this part, we introduce *triple graph* and the way to extract source document information and background knowledge automatically.

2.1 Motivation for triple presentation

Compared to Wiki Page, triple based enrichment helps to reduce noise illustrated in Section 1. Compared to bag of words, triple based presentation help to reduce ambiguity of single word which is shown in

[2] http://catalog.ldc.upenn.edu/LDC2006T06

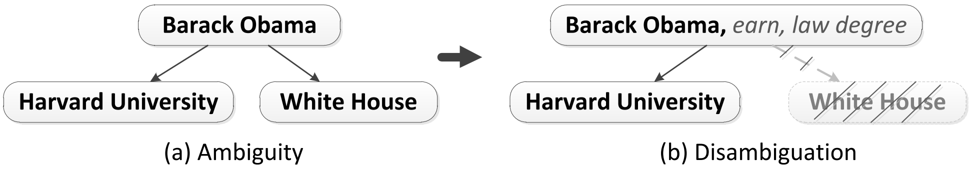


Figure 2: The motivation for the form of triple (a) relevance ambiguity of single word *Obama*, which is related to Harvard and White House (b) disambiguation with the help of other triple elements, where "*earn, law degree*" help to limit *Obama* to the graduate of Harvard.

Figure 2. Figre 2 (a) shows that the single word of *Obama* is related to multiple semantic information such as *Harvard University* as a law graduate and *White House* as the president. After introducing the information from other elements of the triple, "*earn, law degree*" help to disambiguate and limit *Obama* to the law graduate of Harvard University only in Figure 2 (b). The form of triple has been used as the presentation of knowledge in some researches such as knowledge base (Hoffart et al., 2013).

2.2 Nodes in the Graph

There are two kinds of nodes in the triple graph: source document nodes (*sd-nodes*) and background knowledge nodes (*bk-nodes*). Both of them are extracted automatically with *Open Information Extraction (Open IE)* technology which focuses on extracting assertions from massive corpora without a pre-specied vocabulary (Banko et al., 2007). Open IE systems are *unlexicalized*-formed only in terms of syntactic tokens and closed-word classes, instead of specific nouns and verbs at all costs.

There are existing Open IE systems such as TextRunner (Banko et al., 2007), WOE (Wu and Weld, 2010), and StatSnowball (Zhu et al., 2009). The output of these systems has been used to support many NLP tasks such as learning selectional preference (Ritter et al., 2010), acquiring sense knowledge (Lin et al., 2010), and recognizing entailment (Schoenmackers et al., 2010). In this work, we use the famous Open IE system ***Reverb*** (Etzioni et al., 2011), which is generated from TextRunner (Etzioni et al., 2008), to extract source document information and background knowledge automatically. We use the newest version of *ReVerb* (version 1.3) without modification, which is free download on-line [3].

Source document node (*sd-node*) Sd-nodes consists of the information extracted from source document automatically by open information extraction technology (Banko et al., 2007), especially Reverb, the famous Open IE system developed by University of Washington (Etzioni et al., 2011). The output of ReVerb is formed as "$argument_1, predicate, argument_2$", which is naturally presented as triple. In this study, we use ACE corpus as source documents and all sd-nodes are extracted by ReVerb. The setup of automatic extraction makes our method usable in many real applications. To evaluate the effect of automatic extraction, we also use the golden annotation within ACE (Doddington et al., 2004) corpus as source document information and compare the performance that with automatic extraction.

Background knowledge node (*bk-node*) Bk-nodes consist of the background knowledge extracted from external corpus resources automatically by Reverb too. We do not rely on certain existed knowledge base and extract background knowledge from external corpus resources for corresponding source document. This setup makes our methods usable in many real applications. Although we do not rely on special knowledge base, we do adapt our method for the existed knowledge base such as YAGO (Hoffart et al., 2013) and compare the performance to evaluate the effect of different knowledge sources.

2.3 Edges in the Graph

The edges between two nodes indicate their semantic relevance, which is evaluated in Section 3.1. There are two kinds of edges: (1) *sd-node* to *sd-node* (2) *sd-node* to *bk-node*, both of them are undirected. Considering all the relevance score originating from sd-nodes, we connect no edge between *bk-nodes*.

[3] http://reverb.cs.washington.edu/

Edges between sd-nodes All sd-nodes are extracted from the same document, so they should be related to each other. We connect each pair of sd-nodes with an edge and set the weight of edge as their semantic relevance computed in Section 3.1. With this setup, we combine the source document as a whole where different parts affect each other through the edge.

Edges between sd-node and bk-node The basic idea of our model is to propagate relevance score from the sd-nodes to bk-nodes. Hence, we connect each pair of sd-node and bk-node with an edge and set the weight of the edge as their relevance computed in Section 3.1. These edges are all undirected, which indicates that bk-nodes also affect the relevance score of the sd-nodes during the ranking process.

3 Global Ranking Model

In this study, source document *D* is presented as the graph of sd-nodes. For every background knowledge *b*, the task of evaluating the relevance between *b* and *D* is naturally converted into evaluating the relevance between *b* and the graph of sd-nodes. So the relevance between *b* and document *D* can be computed by propagating the *relevance score* from every sd-node of *D* to *b* iteratively. After the convergence, the relevance between *b* and *D* can be evaluated by the *relevance score* of *b*. Intuitively, three factors affect their relevance:

- How many sd-nodes is *b* relevant to ?
- How relevant is *b* to these sd-nodes?
- How important are these sd-nodes ?

For the first factor, *b* should be more relevant to source document *D* if more sd-nodes are relevant to *b*. We capture this information by allowing *b* to receive relevance score from all the sd-nodes. For the second factor, *b* should be more relevant to *D* if more relevant *b* is to sd-nodes. We consider this information by evaluating the relevance between *b* and every sd-node (Section 3.1). For the last factor, important sd-nodes should have higher impact. We consider this information by evaluating the importance of sd-nodes and assigning higher initial value to importance ones (Section 3.3). We combine all factors in the global ranking process to select the top-n relevant background knowledge (Section3.2).

3.1 Relevance Evaluation between Nodes

In this section, we evaluate the semantic relevance between different nodes which is the weight of the edge between them. We introduce *Search Engine* as a resource, which has been proven effective in relevance evaluation (Gligorov et al., 2007). This method is motivated by the phenomenon that the number of results returned by search engine for query $p \cap q$indicates the relevance between p and q.

However, considering the different popularization of queries, this number alone can not accurately express their semantic relevance. For example, query $car \cap automobile$ gets $294,300,000$ results, whereas query $car \cap apple$ gets $683,000,000$, which is 2 times higher than the previous one. Obviously, $automobile$ is more relevant to car rather than $Apple$. The reason of this phenomenon is that $apple$ is far more popular than $automobile$, which increase its possibility of co-occurrence with car. So we consider the number of results for $p \cap q$ together with p and q with $WebJaccard\,Coefficient$ (Bollegala et al., 2007) to evaluate the relevance between p and q according to Formula 1, where $H(p)$, $H(q)$, and $H(p \cap q)$ indicate the number of results for query p, p, and $p \cap q$.

$$WebJaccard(p,q) = \begin{cases} 0 & \text{if } H(p \cap q) \leq C \\ \frac{H(p \cap q)}{H(p)+H(q)-H(p \cap q)} & \text{otherwise.} \end{cases} \quad (1)$$

To convert one "$argument_1, predicate, argument_2$" triple into query, we use $argument_1 \cap argument_2$ as the query for one triple. We have tried $argument_1 \cap predicate \cap argument_2$ which is usually very sparse. Besides, the combination of two arguments usually maintain better semantic completeness of triple compared to other combinations according to our analysis. So this setup aims to balance completeness and sparseness. Accordingly, two triples are combined as $argument_1 \cap argument_2 \cap argument_1^{'} \cap argument_2^{'}$. Considering the scale and noise in the Web data, it is possible for two words to appear together accidentally. To reduce the adverse effects attributed to random co-occurrences, we set 0 to the $WebJaccard\, Coefficient$ of query $p \cap q$, if the number of result is less than a threshold C.

3.2 Iterative Relevance Propagation

Here we propose the relevance propagation based iterative process to evaluate the relevance between certain background knowledge and source document. Note that standard label propagation mainly focuses on classification task (Wang and Zhang, 2008). However, we focus on a ranking problem where the best ranking result is computed during an iterative process in this study. So we make two modifications to suit the ranking problem better: not reseting the relevance score and introducing the propagation between source document information during iteration.

Propagation possibility The edge between $node_i$ and $node_j$ is weighted by $r(i,j)$ to measure their relevance. However, $r(i,j)$ cannot completely present the propagation possibility because one node can be equally relevant to all of its neighbors. Thus, we define $p(i,j)$ based on $r(i,j)$ in formula 2 to indicate the propagation possibility between $node_i$ and $node_j$.

$$p(i,j) = \frac{r(i,j) \times \delta(i,j)}{\sum_{k \in N} r(k,j) \times \delta(k,j)} \tag{2}$$

N is the set of all nodes, $\delta(i,j)$ denotes whether an edge exists between $node_i$ and $node_l$ in the triple-graph or not, which indicates whether they may propagate to each other or not. E is the set of edges.

$$\delta(i,j) = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Iterative propagation There are $n \times n$ pairs of nodes, the $p(i,j)$ of them is stored in a matrix P. we use $\vec{W} = (w_1, w_2, \cdots, w_n)$ to denote the relevance score of all nodes, in which w_i indicates the relevance between $node_i$ and source document D. Here the $node_i$ can indicate both sd-nodes and bk-nodes because they are processed during one fellow step. So that we keep updating both sd-nodes and bk-nodes and do not distinguish them explicitly. The only difference between them is that we initialize the w_i of sd-nodes as its importance to D (Section 3.1) while bk-nodes as 0 at the beginning. We use matrix P together with $\delta(i,j)$ to compute the $\vec{W}$ during a iterative process, where $\vec{W}$ is updated to $\vec{W}'$ during the end of every iteration. The matrix $\vec{W}'$ is updated according to the following Formula 4:

$$\begin{aligned} \vec{W}' &= \vec{W} \times P \\ &= \vec{W} \times \begin{bmatrix} p(1,1) & p(1,2) & \cdots & p(1,n) \\ p(2,1) & p(2,2) & \cdots & p(2,n) \\ \cdots & \cdots & \cdots & \cdots \\ p(n,1) & p(n,2) & \cdots & p(n,n) \end{bmatrix} \end{aligned} \tag{4}$$

each w_i in $\vec{W}$ is updated to $w_i^{'}$ according to the formula 5, where w_i is propagated from all the other $w_j (j \neq i)$ according to their propagation possibility $p(j,i)$. We also introduce the propagation from bk-nodes to sd-nodes, where bk-nodes serve as intermediate to help mining latent semantics.

$$\begin{aligned} w_i^{'} &= w_1 \cdot p(1,i) + w_2 \cdot p(2,i) + \cdots + w_n \cdot p(n,i) \\ &= \sum_{k \in N} w_k \cdot p(k,i) \\ &= \sum_{k \in N} w_k \cdot \left(\frac{r(i,j) \times \delta(i,j)}{\sum_{k \in N} r(k,j) \times \delta(k,j)} \right) \end{aligned} \tag{5}$$

3.3 Importance Evaluation for sd-nodes

The main idea of our model is to propagate relevance score from sd-nodes to bk-nodes (Section 3.2). So the initialization of sd-node is important, which indicates the importance of different source document information. This section solves this problem by evaluating the importance of sd-nodes to source document. We use v_j to denote the initialization of sd-nodes, which indicates the importance of $node_j$ ($node_j \in$ set of sd-nodes) to source document. In this section, we propose a modified relevance propagation method to evaluate v_j for sd-notes. We first construct a triple-graph consisting of sd-nodes only. Then we initialize the relevance score of sd-nodes according to a simple approach based on text frequency (Kohlschütter et al., 2010). We use similar relevance propagation process without resetting the relevance score at the beginning of every iteration, until a global stable state is achieved. Finally, we normalize all the relevance scores to get $\vec{V}$, which indicates the importance of sd-nodes to the source document. We return $\vec{V}$ to the global ranking model (Section 3.2) as part of the input. The initial importance of bk-nodes is set as 0 at the beginning, which denotes that all bk-nodes are ir-relevant to source document before the starting of global ranking process.

4 Experiment

We treat our task as a ranking problem, which takes a document as input and output the ranked list of background knowledge. We evaluate our method as a ranking problem similarly to information retrieval task and focus on the performances of models with different setups.

4.1 Data Preparation

The experiment data consists of two parts: source document information and corresponding background knowledge. To select source documents, we use the *ACE corpus* (Doddington et al., 2004) for 2005 evaluation [4] which consists of 599 articles from multiple sources. We use ReVerb to extract these documents into multi-triples. For background knowledge, we first retrieve relevant web pages with simply term matching method and then extract these pages with ReVerb into a set of triples serving as background knowledge. To ensure the quality, we filter them according to the confidence given by ReVerb.

Besides automatic extraction, we also adapt our system to the golden annotation of ACE as source document information and standard YAGO knowledge base [5] as background knowledge (Hoffart et al., 2013). We compare its performance with that in fully automatic system and evaluate the effect of automatic extraction. For better comparison with YAGO, we retrieve relevant pages from WikiPedia although our automatic extraction method is applicable to any corpus resources.

For every outputted list, three trained annotators check the result and decide which background knowledge is relevant to source document. They work separately and check the same list, so that we can e-valuate their annotation consistence. They totally annotated more than 7000 background knowledge and achieved a $Fleiss'\ Kappa$ value of 0.8066 in best situation and 0.7076 in average between three annotators, which is a good consistence between multi-annotator (Fleiss, 1971). When collision happened, we choose the label selected by more annotators.

4.2 Baseline system

Although we treat our task as a ranking problem, it is difficult to apply corresponding methods in traditional ranking tasks such as information retrieval (IR) (Manning et al., 2008) and entity linking (EL) (Han et al., 2011; Kataria et al., 2011; Sen, 2012) directly in our task. First, both *IR* and *EL* make use of the link structure between web or Wiki pages. However, our task takes single document as input and no link exists between documents which makes it difficult to apply *IR* and *EL* methods such as page rank (Page et al., 1999) and collective method (Han et al., 2011; Sen, 2012) in this task directly. Second, *EL* usually evaluate the text similarity between certain document and target page in WikiPedia. However, our task focuses on the ranking of "$argument_1, predicate, argument_2$" triple, which contains little text information. Lack of text information also limits the application of corresponding methods in our task.

[4] http://catalog.ldc.upenn.edu/LDC2006T06

[5] http://www.mpi-inf.mpg.de/yago-naga/yago

Setup	MAP	P&20
Baseline	0.494	0.377
AutoSD + AutoBK + NoInitial	0.504	0.378
AutoSD + AutoBK + WithInitial	0.531	0.406
GoldSD + AutoBK + NoInitial	0.564	**0.417**
GoldSD + AutoBK + WithInitial	0.553	0.406
GoldSD + YAGO + NoInitial	0.676	0.328
GoldSD + YAGO + WithInitial	**0.676**	0.328

Table 1: The result of our model in different setups: *GoldSD* indicates using annotation of ACE corpus as source document information; *YAGO* indicates using YAGO knowledge base as background knowledge; *AutoSD* and *AutoBK* means aotomatic extraction to source document and background knowledge; *NoInitial* and *WithInitial* means whether using different initial importance to source document information.

For better comparison, we introduce search engine as resource which is proved effective in relevance evaluation (Gligorov et al., 2007) and propose a search engine based strong baseline. As illustrated before, the relevance R_i between background knowledge b_i and source document D has been converted into the relevance between b_i and the triples of D. Hence, we compute R_i by accumulating all r_{ij}, the relevance scores between b_i and every sd-node s_j with the same method in Section 3.1 ($R_i = \sum_{s_j \in S} r_{ij}$, S is the set of sd-nodes). Then we rank all background knowledge according to the value of R_i and output the ranked list as final result. We extract source document and background knowledge automatically in the baseline system, which makes it applicable in different setups.

4.3 Experiment setup

We evaluate our model in different setups. First, we extract both source document information and background knowledge automatically. Second, we use golden annotation of ACE as source document information but extract background knowledge automatically. Third, we use golden annotation of ACE and introduce standard YAGO as background knowledge. For all of them three, we evaluate the different performances with and without initial importance of sd-nodes(Section 3.3). We evaluate the performance with two famous criteria in ranking problem: MAP (Voorhees et al., 2005) requires more accuracy and focuses on the knowledge in higher position; $P\&N$ which require more coverage and pays more attention to the number of relevant ones in $Top\ N$ knowledge. Note that we do not evaluate the $Recall$ performance because there can be millions of background knowledge to be ranked for every document. It is impossible to check all of them. So we focus on the $Top\ N$ candidates and evaluate the performance with MAP and $P\&N$. In this study, we evaluate the Top 20 background knowledge triples which are most easily to be viewed by users.

4.4 Experiment Result

The performance of our model is shown in Table 1. Our search engine based baseline system achieve a rather good performance: a *MAP* value of 0.494 and 0.377 in *P&20*. At the same time, our model outperforms the baseline system in almost every setup and evaluation criterion. The best performance of *MAP* is achieved by *GoldSD+YAGO* (0.676), while the best performance of *P&20* is achieved by *GoldSD+AutoBK* (0.417). To analyze the result further, we find that the initial importance, automatic extraction to source document, and to background knowledge have different effect on the final performance.

4.4.1 Effect of automatic extraction to source document

We use ACE corpus as source documents, which contain golden annotation to document information. So we can evaluate the effect of automatic extraction to source document by comparing the performance with and without golden annotation. The performance without golden annotation is shown in *AutoSD+AutoBK* of Table 1, while the other one shown in *GoldSD+AutoBK*. We can find that the performance of *GoldSD+AutoBK* is better than that of *AutoSD+AutoBK* in both *MAP* and *P&20*, which indicates that golden annotation do help to improve the ranking result.

We further analyze the result and find an interesting phenomenon: these two systems performs greatly different with the setup of *NoInitial*, but equally with the setup of *WithInitial*, which indicates that the performance of *AutoSD+AutoBK* has been improved by evaluating the importance of source document information (Section 3.3). So we can naturally infer that, with a better importance evaluating method in *AutoSD+AutoBK*, we may achieve similar performance compared to that in golden annotation. Note that, *AutoSD+AutoBK* is compatible with any corpus which is more useful in real applications.

4.4.2 Effect of automatic extraction to background knowledge

We evaluate the effect of automatic extraction to background knowledge by comparing the performances between *GoldSD+AutoBK* and *GoldSD+YAGO*. In *GoldSD+AutoBK*, the background knowledge is extracted automatically with ReVerb, which has greater coverage but less accuracy. In contrast, the *GoldSD+YAGO* make use of YAGO as background knowledge, which is less coverage but better accuracy. This difference are reflected on the system performance, where *GoldSD+YAGO* achieves much better result in *MAP*, but much worse in *P&20*. This is partly because that *MAP* focus on the background knowledge in higher position which requires more accuracy, while *P&20* pays more attention to the number of relevant background knowledge which require more coverage.

In general, automatic extraction system has better coverage but less accuracy compared to YAGO based system. However, automatic extraction to background knowledge may help in real applications by improving coverage greatly. Besides, the loss of accuracy is partly due to the technology of information extraction which may be improved in the future. In addition, we can also combine these two ways to acquire background knowledge to balance coverage and accuracy in the future.

4.4.3 Effect of initial importance to source document information

Initial importance to source document information (Section 3.3) is important to the performance of our models as shown in Table 1. The model *AutoSD+AutoBK+WithInitial* outperforms the *AutoSD+AutoBK+NoInitial* compared to other setups, which indicates the help of initial importance to the ranking result. Especially, initial importance to source document information helps most in the setup of *AutoSD+AutoBK*, which is most useful in real applications. So we can naturally infer that, by proposing better importance evaluating method, we may further improve the performance of *AutoSD+AutoBK+WithInitial*, which will great helpful in the future application of this method.

5 Related Work

Document enrichment focuses on introducing external knowledge into source document. There are mainly two kinds of works in this topic according to the resource they relying on. The first line of works make use of *WikiPedia* and enrich source document by linking the entity to its corresponding Wiki page (Bunescu and Pasca, 2006; Cucerzan, 2007). In early stage, most researches rely on the similarity between the context of the mention and the definition of candidate entities by proposing different measuring criteria such as dot product, cosine similarity, KL divergence, Jaccard distance and more complicated ones (Bunescu and Pasca, 2006; Cucerzan, 2007; Zheng et al., 2010; Hoffart et al., 2011; Zhang et al., 2011). However, these methods mainly rely on text similarity but neglect the internal structure between mentions. So another kind of works explore the structure information with collective disambiguation (Kulkarni et al., 2009; Kataria et al., 2011; Sen, 2012; He et al., 2013). These methods make use of structure information within context and resolve different mentions based on the coherence among decisions. Despite the success, the entity linking methods rely on WikiPedia which has great coverage but less accuracy.

Another line of works try to improve the accuracy of enrichment by introducing ontologies (Motta et al., 2000; Passant, 2007; Fodeh et al., 2011; Kumar and Salim, 2012) and structured knowledge such as WordNet (Nastase et al., 2010) and Mesh (Wang and Lim, 2008). In these studies, resources usually provides word or phrase semantic information such as synonym (Sun et al., 2011) and antonym (Sansonnet and Bouchet, 2010). However, these methods rely on special ontologies constructed with supervision or even manually, which is difficult to expand and in turn limits the application of them.

6 Conclusion and Future Work

This study presents a triple based background knowledge ranking model to acquire most relevant background knowledge to certain source document. We first develop a triple graph based document presentation to combine source document together with the background knowledge. Then we propose a global iterative ranking model to acquire $Top\ n$ relevant knowledge, which provide additional information beyond the source document. Note that, both source document information and background knowledge are extracted automatically which is useful in real application. The experiments show that our model achieves better results over a strong baseline, which indicates the effectiveness of our framework.

Another interesting phenomenon is that *YAGO* based enrichment model achieved better ranking accuracy, but less coverage compared to automatic extraction model. To combine these two sources of background knowledge may help to overcome both coverage and accuracy problem. So exploiting proper way to incorporate knowledge base and automatic extraction is an important topic in our future work.

Finally, we believe that this background knowledge based document enriching technology may help in those semantic based NLP applications such as coherence evaluation, coreference resolution and question answering. In our future work, we will explore how to make use of these background knowledge in real applications, hopefully to improve the performance significantly in the future.

Acknowledgements

We thank Muyun Yang and Jianhui Ji for their great help. This work was supported by National Natural Science Foundation of China(NSFC) via grant 61133012, the National 863 Leading Technology Research Project via grant 2012AA011102 and the National Natural Science Foundation of China Surface Project via grant 61273321.

References

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2007. Measuring semantic similarity between words using web search engines. *www*, 7:757–766.

Volha Bryl, Claudio Giuliano, Luciano Serafini, and Kateryna Tymoshenko. 2010. Using background knowledge to support coreference resolution. In *ECAI*, volume 10, pages 759–764.

Razvan C Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, volume 6, pages 9–16.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, volume 7, pages 708–716.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *LREC*. Citeseer.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. 2011. Open information extraction: The second generation. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume One*, pages 3–10. AAAI Press.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Samah Fodeh, Bill Punch, and Pang-Ning Tan. 2011. On ontology-driven document clustering using core semantic features. *Knowledge and information systems*, 28(2):395–421.

Risto Gligorov, Warner ten Kate, Zharko Aleksovski, and Frank van Harmelen. 2007. Using google distance to weight approximate ontology matches. In *Proceedings of the 16th international conference on World Wide Web*, pages 767–776. ACM.

Xianpei Han, Le Sun, and Jun Zhao. 2011. Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 765–774. ACM.

Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. 2013. Learning entity representation for entity disambiguation. *Proc. ACL2013*.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics.

Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. Yago2: a spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61.

Xiaohua Hu, Xiaodan Zhang, Caimei Lu, Eun K Park, and Xiaohua Zhou. 2009. Exploiting wikipedia as external knowledge for document clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 389–396. ACM.

Saurabh S Kataria, Krishnan S Kumar, Rajeev R Rastogi, Prithviraj Sen, and Srinivasan H Sengamedu. 2011. Entity disambiguation with hierarchical topic models. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1037–1045. ACM.

Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. 2010. Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 441–450. ACM.

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. 2009. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 457–466. ACM.

Yogan Jaya Kumar and Naomie Salim. 2012. Automatic multi document summarization approaches. *Journal of Computer Science*, 8(1).

Thomas Lin, Oren Etzioni, et al. 2010. Identifying functional relations in web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1266–1276. Association for Computational Linguistics.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge.

Enrico Motta, Simon Buckingham Shum, and John Domingue. 2000. Ontology-driven document enrichment: principles, tools and applications. *International Journal of Human-Computer Studies*, 52(6):1071–1109.

Vivi Nastase, Michael Strube, Benjamin Börschinger, Cäcilia Zirn, and Anas Elghafari. 2010. Wikinet: A very large scale multi-lingual concept network. In *LREC*.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web.

Patrick Pantel and Ariel Fuxman. 2011. Jigs and lures: Associating web queries with structured entities. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 83–92. Association for Computational Linguistics.

Alexandre Passant. 2007. Using ontologies to strengthen folksonomies and enrich information retrieval in weblogs. In *Proceedings of International Conference on Weblogs and Social Media*.

Alan Ritter, Oren Etzioni, et al. 2010. A latent dirichlet allocation method for selectional preferences. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 424–434. Association for Computational Linguistics.

Jean-Paul Sansonnet and François Bouchet. 2010. Extraction of agent psychological behaviors from glosses of wordnet personality adjectives. In *Proc. of the 8th European Workshop on Multi-Agent Systems (EUMAS10)*.

Stefan Schoenmackers, Oren Etzioni, Daniel S Weld, and Jesse Davis. 2010. Learning first-order horn clauses from web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1088–1098. Association for Computational Linguistics.

Prithviraj Sen. 2012. Collective context-aware topic models for entity disambiguation. In *Proceedings of the 21st international conference on World Wide Web*, pages 729–738. ACM.

Koun-Tem Sun, Yueh-Min Huang, and Ming-Chi Liu. 2011. A wordnet-based near-synonyms and similar-looking word learning system. *Educational Technology & Society*, 14(1):121–134.

Ellen M Voorhees, Donna K Harman, et al. 2005. *TREC: Experiment and evaluation in information retrieval*, volume 63. MIT press Cambridge.

Xudong Wang and Azman O Lim. 2008. Ieee 802.11 s wireless mesh networks: Framework and challenges. *Ad Hoc Networks*, 6(6):970–984.

Fei Wang and Changshui Zhang. 2008. Label propagation through linear neighborhoods. *Knowledge and Data Engineering, IEEE Transactions on*, 20(1):55–67.

Fei Wu and Daniel S Weld. 2010. Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127. Association for Computational Linguistics.

Wei Zhang, Yan Chuan Sim, Jian Su, and Chew Lim Tan. 2011. Entity linking with effective acronym expansion, instance selection and topic modeling. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 1909–1914. AAAI Press.

Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu. 2010. 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*, pages 483–491. Association for Computational Linguistics.

Jun Zhu, Zaiqing Nie, Xiaojiang Liu, Bo Zhang, and Ji-Rong Wen. 2009. Statsnowball: a statistical approach to extracting entity relationships. In *Proceedings of the 18th international conference on World wide web*, pages 101–110. ACM.