Passage Retrieval for Information Extraction using Distant Supervision

Wei Xu° Ralph Grishman° Le Zhao*

°New York University
New York, NY, USA
xuwei,grishman@cs.nyu.edu

*Carnegie Mellon University
Pittsburgh, PA, USA
lezhao@cs.cmu.edu

Abstract

In this paper, we propose a keyword-based passage retrieval algorithm for information extraction, trained by distant supervision. Our goal is to be able to extract attributes of people and organizations more quickly and accurately by first ranking all the potentially relevant passages according to their likelihood of containing the answer and then performing a traditional deeper, slower analysis of individual passages. Using Freebase as our source of known relation instances and Wikipedia as our text source, we collected a weighted set of keywords indicative of each relation and then use it to re-rank the passages retrieved by the Lemur search engine. Experiments show that our algorithm significantly outperforms state-of-the-art passage retrieval techniques in evaluations of both individual passage retrieval and end-to-end information extraction.

1 Introduction

Large-corpus information extraction involves the extraction of pre-specified types of relations and events from large corpora. For example, the Knowledge Base Population (KBP) slot-filling task (Ji et al., 2010) involves finding, from a large corpus, a few dozen attributes of a specified person or organization.

In many cases we do not have the time to perform in-depth extraction for all attributes over the entire corpus. Consequently, addressing this task typically involves a blend of traditional question answering (QA) and information extraction (IE) methods. Like QA, we need to begin with passage retrieval, where a passage can range from a sentence to a piece of text or a document. However, unlike QA, we have a fixed inventory of relations and a fixed set of expected answer types (e.g. employer of a person). This allows us to bring to bear the more specialized learning methods of IE to tune the passage retrieval for each relation of interest.

To the best of our knowledge, we are the first to systematically study the passage retrieval algorithm for information extraction and propose a novel distant supervision approach to obtain a list of weighted keywords for each relation. Distant supervision (Mintz et al., 2009) makes use of noisy training data generated automatically from a related, but different, type dataset to solve problems on another type of data. Instead of a handful of human-selected keywords, we automatically learn hundreds or thousands of indicative keywords from a freely available online resource, Freebase, which is similar to Wikipedia Infoboxes. Passages are ranked and retrieved based on these keywords indicative of certain relations. We then feed individual passages to a traditional IE system or to an answer extraction component as used in QA systems to obtain the final outputs. Both the training and testing procedures of our method require only statistics of surface words and named entities in the text and thus are time efficient.

This paper addresses the following questions:

1) How can we tune passage retrieval for a particular relation?

2) How do distant learning methods apply to the passage retrieval task?

3) How much do these methods improve over typical QA passage retrieval?

We will measure the improvement in two ways:

1) ability to find a relevant passage, such as reduction in the number of passages the system must examine and increase in the proportion of relevant passages in top-ranked ones;

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 1046–1054,
Chiang Mai, Thailand, November 8 – 13, 2011.

2) improvement in the precision and recall of information extraction by taking passage relevance into account.

2 Previous Work

Relatively little work has been done to investigate in detail the quality of the IR for large-corpus IE and take advantage of the more constrained relations of interest compared to traditional QA. The Knowledge Base Population (KBP) track at TAC 2010 (Ji et al., 2010) evaluates the ability of automated systems to discover information about named entities. Its slot-filling task is to find answers to queries asking a few dozen attributes of a specified entity, such as the 'employee_of' attribute of a given PERSON entity. We refer to the given entity as the *target entity*, and the attributes of entities as *slot types*. In past KBP competitions, many participants (Li et al., 2009; Byrne and Dunnion, 2010; Chen et al., 2010) exploited a QA system to fill slots by constructing queries based on target entities and slot types. However, their query templates contain only a few additional query terms other than the target entity name, which are mostly obtained manually.

Most of QA systems use the question words as-is or with expansion to form the retrieval system query. Various query expansion approaches have been used to tackle the passage-query mismatch problem, including relevance feedback (Derczynski et al., 2008), ontologies (Bhogal et al., 2007), semantic lexica (Ofoghi et al., 2006), etc. As a data-driven approach, relevance feedback is sensitive to the quality of first time retrieval. Our use of Freebase, a freely available large semantic database, to provide distant supervision requires neither labeled data nor costly constructed knowledge models.

Some researchers (Grishman and Min, 2010; Chrupala et al., 2010; Surdeanu et al., 2010) integrated IR and IE together. Surdeanu et al. (2010) coupled the entity name with a handful of hand-selected trigger words for each slot type as queries to IR system in an effort to boost the ranking of sentences likely to contain the relations of interest. Chrupala et al. (2010) proposed one of the most customized passage retrieval components for large-corpus IE. Besides the target name entity, they take into account the type of expected named entity (such as ORGANIZATION for the 'employee_of' relation) and expand queries by predefined words that are predictive for specific slot types (such as 'work' for the 'employee_of' relation). There are also relevant works emerging from the IR community in the Related entity Finding (REF) task in the TREC Entity Track (Balog et al., 2010), which is to return a ranked list of related entities given an expected type of entity and a brief description (query) of the relation in free text. Fang et al. (2010) ranked entities by their relevance to the query at the document, passage and entity level, primarily based on the similarity between terms.

In all this previous work, the limited number of query terms has become the performance bottleneck of the passage retrieval for large-corpus information extraction.

Perhaps most similar to our distant supervision keyword learning approach for passage retrieval is the semi-automatic method of Nguyen et al. (2007), who extract only several keywords for each relation from Wikipedia and study only the dependency subtrees that contain those keywords. In contrast to their tf-idf model followed by a manual selection step, our algorithm allows us to fully automatically extract hundreds or even thousands of keywords with a weight indicating their relevance to each relation.

Mintz et al. (2009) proposed a distant supervision approach for relation extraction using a rich-featured logistic regression model. Like us, they used Freebase as a source of known relation instances and Wikipedia as a text source to create noisy training data and tested on the Wikipedia data. Our approach differs from theirs in several ways. First, our main concern is the speed required for large-corpus IE and reducing the amount of text to process by passage retrieval, while they use deep NLP features such as parsing and process the whole corpus. Second, we assure the quality of output by using a supervised information extraction system trained on golden data, while their performance is constrained by noisy training data. Third, we evaluate on a corpus that consists of news and web data, while they test on Wikipedia data that is from the same source as the training data. We prove that our method is adaptive to new domains because it is based on lexical statistics and thus tolerant to noise in the training data.

3 Freebase and Wikipedia

Freebase[1] is a freely available online database of structured knowledge. It collects information about approximately 20 million entities (such as

[1] http://www.freebase.com

people, places, books, etc.) from a wide variety of sources, including Wikipedia, MusicBrainz (music), NNDB (biographical information), user editing, etc.

Following the literature (Mintz et al., 2009), we refer to each attribute of a person or organizations as an ordered, binary 'relation' between entities. We refer to individual entity pairs in a relation as relation instances. For example, the 'employee_of' relation indicates a person's employment history with zero to many companies, of which an instance can be <Steve Jobs, employee_of , Apple Inc.>.

Freebase provides us a set of relations and entity pairs that participate in those relations. Often understood as a Wikipedia-turned-database, Freebase also distinguishes similar names and includes exact wiki articles for many entities, and thus forms a perfect source for training texts. We will later discuss the effectiveness and adaptivity of using Wikipedia as training corpus at the end of Section 4 and in Section 7.

4 Learning Indicative Keywords for Relations from Freebase

Our intuition in this research is that there exist some keywords that provide clues to the text passages containing the relationship. For example, a sentence or a couple of successive sentences which contain words like 'hire', 'work', 'appoint', and so on, are highly likely to express the 'employee_of' relation.

We identify such keywords using a distant supervision approach. First, we obtain entity pairs for a certain relation and the Wiki articles of these entities from Freebase. Then we locate the entities in the training corpus to collect sample sentences for each relationship. Finally, we rank the words based on their frequency in those sample sentences versus all sentences in the training corpus.

Like previous work (Nguyen et al., 2007; Mintz et al., 2009), our distant supervision assumption is that if two entities participate in a relation, a sentence that contains those two entities might express that relation. For our training corpus, we choose Wikipedia in this paper since it well supports Freebase data and is believed to have high grammatical correctness compared to that of the web overall. It is also feasible to use another corpus as long as it potentially contains text about the relationships of interest and the known entity pairs.

Unlike (Mintz et al., 2009), which considers any sentences containing the pair of entities, we take advantage of the fact that Wikipedia records only one article per language for a real world entity. Any sentences in the person's Wiki article that contains the person name or pronouns ('she' or 'he') and his/her employer name are considered positive examples, others as negative examples. This setting can capture more true positive examples. What is more, Freebase includes name variants of a real world entity and thus gives a better chance to match the person name back to his/her Wiki pages.

We then determine if a word is indicative and to what degree for a certain type of relation based on its occurrence in positive and negative examples. We use the morphological base forms to replace their inflectional variants in the process. The indicative score I of word w for relation R is calculated by the following formula:

$$I(w,R) = \frac{Pos}{Pos+Neg} \times \frac{Pos}{Pos+\alpha} \qquad (1)$$

where $Pos = |S_{pos}(w,R)|$ and $Neg = |S_{neg}(w,R)|$, $S(w,R)$ are all the positive/negative sentences for relation R which contain word w. The larger the weight the more likely the word indicates the relationship. The first factor $Pos/(Pos+Neg)$ favors words which are more frequent in positive sentences versus all sentences. The factor $Pos/(Pos+\alpha)$ is used to reduce low frequency word noise, where α is a constant to be set experimentally. All the words with $Pos > Neg$ are extracted to form a weighted keyword list for each relation.

Table 1 shows the top-weighted keywords learned for the 'employee_of' and 'member_of' relations, using the January 2011 data dump of Freebase. The Freebase contains 10702 instances of 'employee_of' relation (named '/business/ employment_tenure/' in Freebase), among which 4497 are found in its corresponding Wikipedia articles. In total, 6574 positive, 93756 negative examples and a weighted set of 2436 keywords indicative of the relation are extracted. For 'member_of' relation (named '/organization/ organization_membership/' in Freebase), there are 586 instances in Freebase, 244 positive and 13749 negative examples extracted from Wikipedia pages. With many fewer training examples provided by Freebase for the 'member_of' relation, only 290 keywords are learned, but they proved to be effective in the experiments in Section 7.

employee_of		
1-10	**11-20**	**21-30**
faculty	currently	headquarter
professor	chairman	since
emeritus	join	housemaster
chancellor	founder	dynamo
teach	chief	yesterday
executive	department	retail
dean	conglomerate	officer
rector	head	merger
lecturer	subsidiary	director
therapeutics	company	president
31-40	**41-50**	**51-60**
shareholder	appointment	appoint
at	physiology	instructor
chain	professorship	merge
position	retailer	zoology
supermarket	owner	former
psychology	until	current
creative	acquisition	studio
associate	holding	adjunct
chair	developer	found
teaching	assistant	vice
member_of		
1-5	**6-10**	**11-15**
fraternity	gradual	president
sorority	elect	peaceful
fraternal	guerrilla	founder
member	carve	fellow
membership	hence	sector

Table 1. List of top-weighted keywords

It is very interesting that many keywords found may not be intuitive, such as "currently", "until", and "supermarket" for the 'employee_of' relation. Though they do not directly imply the occurrence of the relations, they tend to co-occur with the relations, thus helpful in retrieval. Moreover, our system does not rely on only one keyword to make decisions and thus is robust. We also notice that the 'member_of' relation in Freebase has a bias towards the education domain; however, this has little impact in general performance, as shown in Section 7, because the keyword set still covers the most common keywords such as 'member' and 'founder' while the more domain-specific words are rarely seen in the text. This is likely because usually the queried entity helps in disambiguating the relation words and focusing on the right subset of the relation words.

5 Passage Retrieval for Information Extraction

5.1 Lemur

Our goal is to develop a speedy passage retrieval model for the IE task without using complex NLP techniques.

We first employ the high-performance Lemur passage retrieval engine to select relevant passages about the target named entity from a large corpus. Lemur (Metzler and Croft, 2004) implements a model combining language modeling and an inference network and has been widely used in large-corpus IE and QA systems. We define a passage as a natural paragraph in the texts and use Lemur's default setting, which shows a satisfactory capability to retrieve most passages containing a given entity in experiments.

Then, we re-rank the retrieved passages in descending order of their probability of containing the target relation R (e.g. 'employee_of').

5.2 Baseline

Our baseline passage retrieval algorithm for information extraction is derived from the state-of-the-art methods used in the IE community (Chrupala et al., 2010; Surdeanu et al., 2010), IR community (Zhao and Callan, 2008) and Question Answering community (Tellex et al., 2003; Hickl et al., 2007; Moldovan et al., 2007; Gómez et al., 2007). The baseline (and our proposed approach) are intended to involve minimal human-constructed knowledge, annotated data and complex NLP processing.

In the baseline method, the ranking score B of a passage P, with respect to relation R and target entity E, consists of four elements:

$$B(P,R,E) = W_1(P,E) + W_2(P,R) + W_3(P,R) + W_4(P,E) \quad (2)$$

Please note that the constants c in each scoring functions $\boldsymbol{W}$ are only used to define cascading criteria, e.g. any passage that contains the target named entity will be ranked higher than those do not, and thus can be set rather arbitrarily.

1) **The target named entity ($\boldsymbol{W_1}$):** We ensure that passages that (partly) include the string of entity names rank higher than those only containing pronouns.

$$W_1(P,E) = \begin{cases} c_{1a}, if\ P\ contains\ E \\ c_{1b}, if\ P\ partly\ contains\ E \\ c_{1c}, if\ P\ contains\ pronouns \\ 0,\ otherwise \end{cases} \quad (3)$$

where $c_{1a} > c_{1b} > c_{1c}$ are arbitrary constants larger than any possible value of $W_2 + W_3 + W_4$. The passage partly contains the entity when it shares at least a word in common with the entity name.

2) **The named entity of the expected type (W_2):** The named entity of the type that is to be found for the relation is called the expected named entity, (e.g. 'ORGANIZATION' for relation 'employee_of'). We run the Stanford Named Entity Recognizer (Finkel et al., 2005) on Lemur's passage retrieval outputs, preferring passages that contain a named entity of the sought type, and more strongly preferring names that have not appeared in previously retrieved passages (novel names).

$$W_2(P,R) = \begin{cases} c_2 + c_3, if\ P\ contains\ novel\ expected\ entity \\ c_3, if\ P\ contains\ expected\ entity \\ 0, otherwise \end{cases} \quad (4)$$

where c_2 and c_3 are arbitrary constants larger than any possible value of $W_3 + W_4$.

3) **The expansion terms (W_3):** The expansion terms include predefined words that are predictive of or related to a specific relation. We adapt the indicative word list used by Surdeanu et al. (2010) in their KBP system, which include several words for each relation. We also use a list containing 635 common title words as related terms for the 'employee_of' relation. This effort is to simulate the usage of WordNet and ontologies for term expansions but is considered to be more accurate and comprehensive since these terms are collected for the purpose of these particular relations.

$$W_3(P,R) = \begin{cases} c_4, if\ P\ contains\ related\ terms\ of\ R \\ 0, otherwise \end{cases} \quad (5)$$

where c_4 is a arbitrary constant set not less than 1 (thus W_4).

4) **The original rank in Lemur (W_4):** We also take into account the rank from Lemur's passage retrieval.

$$W_4(P,E) = \frac{1}{LemurRank(P,E)} \quad (6)$$

where $LemurRank(P,E)$ is the rank of the passage P as returned from Lemur when querying for entity name E.

5.3 Passage Retrieval using Indicative Keywords Learned from Freebase (IKFB)

As in the baseline, we give top priority to the passages that contain the target named entity and at least a named entity of the expected type for the relation. We discard other weighting schema in Section 5.2 but apply our learned list of weighted indicative keywords instead.

$$K(P,R,E) = W_1(P,E) + W_2(P,R) + W_5(P,R) \quad (7)$$

5) The indicative keywords (W_5):

$$W_5(P,R) = \sum_{w \in L_{distant}(R) \cap w \in Top_n(P)} I(w,R) \quad (8)$$

where $L_{distant}(R)$ is the set of indicative keywords obtained by the approach described in Section 4, $Top_n(P)$ is the set of top n words weighted by $I(w,R)$ which appears in passage P. The number n is experimentally set as 5 in this paper; it serves to prevent preferring longer passages while decreasing the relevance of extremely short passages.

6 Evaluation Issues

6.1 Test Data

In this paper, we use the KBP corpus as test data, which includes about 1.3 million documents of newswire and 0.5 million of web data.

We evaluate passage retrieval algorithms on one of the most frequent relations in the KBP task (Ji et al., 2010; Chen et al., 2010): 'employee_of'. We also experiment on 'member_of' relation, but due to limitations of space we only present MRR values for this relation. The KBP 2010 training data prepared by the Linguistic Data Consortium and by the participants, including 67 person entities, 54 instances each for 'employee_of' and 'member_of' relation, are used as test data in our experiments. Please note that one entity may be involved in multiple relation instances, e.g. a person may have multiple employers.

6.2 Evaluation Metrics

We evaluate the passage retrieval algorithms in two ways. First, we measure the performance of passage retrieval as an independent system, in the context of IR and QA. Second, we examine its impact on the end-to-end information extraction pipeline, in the context of IE.

6.2.1 Evaluation of Individual PR system

Following the literature of passage retrieval (Gómez et al., 2007; Roberts and Gaizauskas, 2004) and question answering, we use three metrics in the experiments, *Coverage*, *Redundancy* and *Mean Reciprocal Rank (MRR)*.

Let Q be the relation query set, D all possible passages which are retrieved by Lemur, $A_{D,q}$ the subset of D which contains correct answers for $q \in Q$, and $F^S_{D,q,n}$ the n top-ranked passages in D retrieved by a retrieval system S responding to query q.

The *coverage* of a retrieval system S for a query set Q and passage collection D at rank n is defined as:

$$coverage^S(Q,D,n) \equiv \frac{\left|\{q \in Q | F^S_{D,q,n} \cap A_{D,q} \neq \emptyset\}\right|}{|Q|} \quad (9)$$

The *answer redundancy* (or simply redundancy) is defined as:

$$redundancy^S(Q,D,n) \equiv \frac{\sum_{q \in Q}\left|F^S_{D,q,n} \cap A_{D,q}\right|}{|Q|} \quad (10)$$

The *Mean Reciprocal Rank (MRR)* (Voorhees, 1999) is defined as:

$$mrr^S(Q,D,n) \equiv \frac{\sum_{q \in Q} rr\left(q, F^S_{D,q,n}\right)}{|Q|} \quad (11)$$

where rr is the Reciprocal Rank which is the inverse of the rank of the first returned passage which contains the answer; or 0 if the answer is not found in the top n retrieved passages. The function is defined as:

$$rr\left(q, F^S_{D,q,n}\right) \equiv \begin{cases} {}^1/_k, if\ \exists k: k = \underset{1 \le i \le n}{\operatorname{argmin}}\left(F^S_{D,q,i} \cap A_{D,q} \neq \emptyset\right) \\ 0, otherwise \end{cases} \quad (12)$$

The coverage gives the proportion of relation instances (correct answers to the query) that can be found within the top n passages retrieved for each query. The redundancy gives the average number, per question, of passages within the top n ranks retrieved which contain a correct answer. The MRR is the average of the reciprocal ranks of the first ranked passage containing a correct answer.

6.2.2 Evaluation of End-to-End IE system

We also evaluate the impact of the passage retrieval algorithm on the final output of a large-corpus information extraction system.

6.2.2.1 Traditional IE Pipeline

We exploit a simple two-stage pipeline architecture for the KBP task. First, we retrieve passages related to the target entity. Then we apply to those passages a traditional information extraction system (Grishman et al., 2005; Ji and Grishman, 2008) to extract relations, which was originally created for the NIST Automatic Content Extraction (ACE) Evaluations. Its relation extraction component uses maximum entropy models, incorporating diverse lexical, syntactic, semantic and ontological knowledge.

To generate the final results, there could be different strategies. We use a simple strategy in this paper, which suffices to show the capability of our system, outputting the answers of the m top-ranked passages that provide an answer (possibly duplicate). A more delicate design is not a focus of this research.

6.2.2.2 QA-like Pipeline

The new passage retrieval IKFB system also allows us to create a QA-like pipeline for large-scale information extraction. Besides applying a sophisticated IE system to the retrieved passages using deep NLP techniques, such as coreference resolution, we can exploit answer extraction/selection components similar to many QA systems. Some common answer extraction/selection approaches, e.g. using distance from keywords, can possibly boost the speed by avoiding deep analysis and improve recall/precision by finding answers that the traditional IE system misses.

Consider the target entity "John Dewey" for example; the IE system failed to extract any employment information from the corpus. Our IKFB algorithm assigns top rank to the following passage:

"An institution that sees itself as an unconventional alternative to other colleges, the New School was founded in 1919 by a group of professors, including the philosopher and education reformer John Dewey, who had resigned in protest from Columbia. They could not abide by a stance taken by Columbia's president at the time, Nicholas Murray Butler, that faculty members had to support America's entry into World War I."

We investigate the potential value of passage ranking by implementing a simple answer extraction component and using its output when the IE pipeline failed to provide any answers to the probe query. The primitive method is to find the organization name that is closest to the target entity in the first top ranked passage, e.g. "Columbia" for the above passage (note: "New School" is missed by the Stanford named entity tagger). A distance-based answer extraction component is good at dealing with complicated language phenomena, since it is less likely to face data sparsity problem or syntactic analysis errors than many IE approaches do.

6.2.2.3 Measurement Metrics

In this framework, we use the traditional measures for evaluating IE, *precision* and *recall*. Following the symbols defined in section 6.2.1, *precision* and *recall* are defined as:

$$precision^S(Q,D,m) \equiv \frac{\sum_{q\in Q}|T^S_{D,q,m} \cap B_{D,q}|}{\sum_{q\in Q}|T^S_{D,q,m}|} \quad (14)$$

$$recall^S(Q,D,m) \equiv \frac{\sum_{q\in Q}|T^S_{D,q,m} \cap B_{D,q}|}{\sum_{q\in Q}|B_{D,q}|} \quad (15)$$

where m is as mentioned in Section 6.2.2.1 about the output generating strategies, $T^S_{D,q,m}$ are the system output, and $B_{D,q}$ are the golden answers.

7 Experiment Results

As we described in Section 6, we carry out experiments for the "employee_of" relation on the training data from KBP 2010, which includes 67 entities and 54 instances of the relation (involving 30 entities) as keys. Using the January 2011 data dump of Freebase as our sources of known relation instances and Wikipedia as out text source, we collect a weighted set of keywords indicative of the relation and then use it to re-rank the passages retrieved by the Lemur engine.

Three passage retrieval algorithms are compared:

- **Lemur**: the passage retrieval functionality provided by Lemur using only the target entity name. An example query looks like this: #combine[p](#5(John Dewey).
- **Baseline**: a baseline approach that prioritizes the passages retrieved by Lemur which contain the expected answer type, relevant terms of the relation, etc.
- **IKFB**: our proposed approach using the weighted keywords learned by distant supervision on Freebase data.

The constants in the ranking formulas were set as follows: $c_{1a} = 30000$, $c_{1b} = 20000$, $c_{1c} = 10000$, $c_2 = 1000$, $c_3 = 100$, $c_4 = 1$. Other values of these constants would be equally effective, as long as they distinguish the priority of each scoring criteria.

7.1 Performance of Individual Passage Retrieval

In Figure 1, we can see the improvement of our passage retrieval algorithm for information extraction usage with respect to the Lemur search engine and baseline system.

The coverage gives the proportion of correct answers that can be found within the top n passages retrieved for each query, if using a perfect information extraction system. For 31.5% of relation instances, only one passage had to be examined to retrieve the employment information using IKFB, while this number is 16.7% for the baseline system, 9.3% for Lemur. The coverage of the baseline and IKFB converge as the number

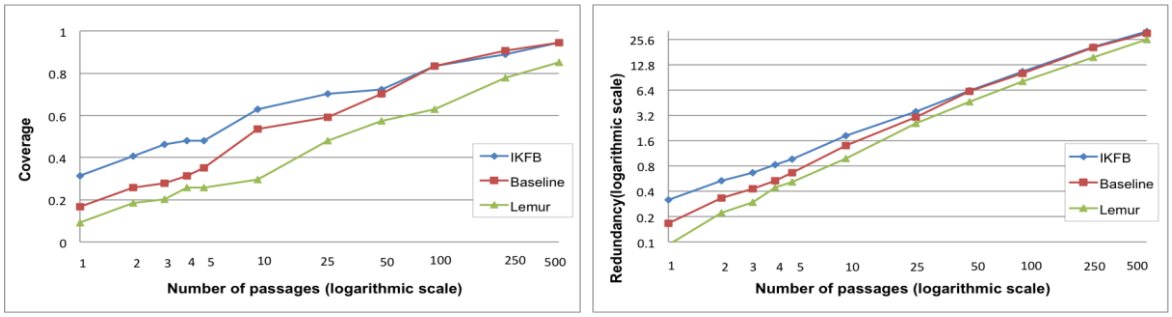

Figure 1. Performance of Individual Passage Retrieval Systems

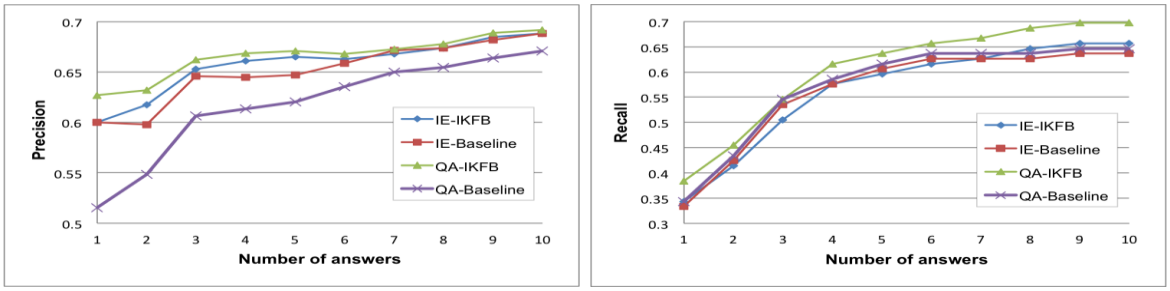

Figure 2. Performance of End-to-End IE
extracting answers for top-ranked passages for each target entity

of passages retrieved increases.

The redundancy gives the average number of passages within the top n retrieved passages which contain a correct answer. On average 18.7% of the top 10 ranked passages by IKFB contain answers, while the number is 5.93% for baseline system.

Table 2 represents the Mean Reciprocal Rank of all three systems and evaluations for all passages ($n = \infty$ in Formula 11). In the previous figure, we can appreciate that the difference in both coverage and redundancy decreases with the number of passages.

System	employee_of	member_of
IKFB	0.409	0.260
Baseline	0.269	0.149
Lemur	0.180	0.079
p-value	0.0092	0.0044

Table 2. Comparision of MRR
p-value: comparing IKFB to Baseline

The paired, two-tailed Student's t test (Smucker et al., 2007) shows that our proposed algorithm IKFB is significantly superior to the baseline in terms of MRR.

7.2 Impact of Passage Retrieval on Information Extraction

Figure 2 represents the performance of an end-to-end information extraction system that extracts the top m ($m = 1{\sim}10$) answers using different passage retrieval algorithms. Both QA and IE pipelines are shown.

Because the information in the training data is incomplete, the output answers were examined manually; another 45 relation instances were discovered besides the given 54 keys in KBP data. IKFB achieves somewhat higher precision with similar recall. It ranks the passages containing the answer higher, while ranking the passages containing the correct answers ahead of those which may suggest wrong answers to the IE system.

There is great room for improvement on answer extraction, such as using a QA-pipeline. It is not uncommon that the information extraction system fails to extract the right answers even when the passages containing them are retrieved and ranked at top. The IE system can only successfully locate 7 out of the 54 given keys in 2.4% of the top 10 ranked passages by IKFB, although 17 keys are contained in 18.7% of these retrieved passages.

Of the total 67 person entities in our test data, the IE-pipeline is not able to extract any employment information for 12 of them. However, using the primitive QA-pipeline, we are able to recover 4 of them while introducing 6 new errors. As shown in Figure 2, the integration of the QA-pipeline to the IE-pipeline improves the end-to-end system performance by 3-5%, since low recall is the most crucial problem of traditional IE systems. Good ranking is particularly important to the IE+QA pipeline; as Figure 2 shows, adding QA to the baseline produces a considerable loss of precision.

8 Conclusions and Future Work

We have presented a novel method to extract keywords from Wikipedia articles and rank passages for information extraction. The key features of our method includes: (1) combining the advantages of question answering and information extraction techniques; (2) involving no complicated or time consuming processes; (3) requiring no costly annotation but making use of freely available Wikipedia and Freebase.

Given the success of our primitive QA-pipeline, we plan to implement a more competitive and customized question answering system for information extraction tasks. We also consider looking more deeply into the adaptiveness from Wikipedia to texts from other sources in order to further improve the quality of weighted keywords.

Acknowledgments

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory (AFRL) contract number FA8650-10-C-7058. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, AFRL, or the U.S. Government.

We wish to thank Adam Meyers and Satoshi Sekine of New York University and Heng Ji of the City University of New York for their advice.

References

Krisztian Balog, Pavel Serdyukov and Arjen P. de Vries. 2010. Overview of the TREC 2010 Entity Track. In *TREC 2010*.

J. Bhogal, A. Macfarlane and P. Smith. 2007. A review of Ontology Based Query Expansion. *Information Processing and Management*, 43(4), 866-886.

Lorna Byrne and John Dunnion. 2010. *UCD IIRG at TAC 2010 KBP Slot Filling Task*. In *TAC 2010 Workshop*.

Zheng Chen, Suzanne Tamang, Adam Lee, Xiang Li, Wen-Pin Lin, Matthew Snover, Javier Artiles, Marissa Passantino, Heng Ji. 2010. CUNY-BLENDER TAC-KBP2010 Entity Linking and Slot Filling System Description. In *TAC 2010 Workshop*.

Grzegorz Chrupala, Saeedeh Momtazi, Michael Wiegand, Stefan Kazalski, Fang Xu, Benjamin Rogh, Alexandra Balahur and Dietrich Klakow. 2010. Saarland University Spoken Language Systems at the Slot Filling Task of TAC KBP 2010. In *TAC 2010 Workshop*.

Leon Derczynski, Jun Wang, Robert Gaizauskas and Mark A. Greenwood. 2008. A Data Driven Approach to Query Expansion in Question Answering. In *COLING 2008 Workshop on Information Retrieval for Question Answering*.

Yi Fang, Luo Si, Zhengtao Yu, Yantuan Xian and Yangbo Xu. 2009. Entity Retrieval by Hierarchical Relevance Model, Exploiting the Structure of Tables and Learning Homepage Classifiers. In *TREC 2009*.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *ACL 2005*.

José Manuel Gómez, Paolo Rosso and Emilio Sanchis. 2007. Re-ranking of Yahoo Snippets with the JIRS Passage Retrieval System. In *IJCAI 2007 Workshop on Cross Lingual Information Access*.

Ralph Grishman and Bonan Min. 2010. New York University KBP 2010 Slot-Filling System. In *TAC 2010 Workshop*.

Ralph Grishman, David Westbrook and Adam Meyers. 2005. NYU's English ACE 2005 System Description. In *ACE 2005 Workshop*.

Andrew Hickl, Kirk Roberts, Bryan Rink, Jeremy Bensley, Tobias Jungen, Ying Shi and Johan Williams. 2007. Question Answering with LCC's CHAUCER-2 at TREC 2007. In *TREC 2007 Workshop*.

Heng Ji and Ralph Grishman. 2008. Refining Event Extraction Through Cross-document Inference. In *ACL 2008*.

Heng Ji, Ralph Grishman, H.T. Dang, K. Griffitt and J. Ellis. 2010. Overview of the TAC 2010 Knowledge Base Population Track. In *TAC 2010 Workshop*.

Fangtao Li, Zhicheng Zheng, Fan Bu, Yang Tang, Xiaoyan Zhu and Minlie Huang. 2009. THU QUANTA at TAC 2009 KBP and RTE Track. In *TAC 2009 Workshop*.

Donald Metzler and W. Bruce Croft. 2004. Combining the Language Model and Inference Network Approaches to Retrieval. *Information Processing and Management Special Issue on Bayesian Networds and Information Retrieval*, 40(5), 735-750.

Mike Mintz, Steven Bills, Rion Snow and Dan Jurafsky. 2009. Distant Supervision for Relation Extraction without Labeled Data. In *ACL 2009*.

Dan Moldovan, Christine Clark and Mitchell Bowden. 2007. Lymba's Power Answer 4 in TREC 2007. In *TREC 2007 Workshop*.

Dat P.T. Nguyen, Yutaka Matsuo and Mitsuru Ishizuka. 2007. Relation Extraction from Wikipedia Using Subtree Mining. In *AAAI 2007*.

Bahadorreza Ofoghi, John Yearwood and Ranadhir Ghoshi. 2006. A Semantic Approach to Boost Passage Retrieval Effectiveness for Question Answering. In *ASCS 2006*.

Ian Roberts and Robert Gaizauskas. 2004. Evaluating Passage Retrieval Approaches for Question Answering. In *ECIR 2004*.

Mark D. Smucker, James Allan and Ben Carterette. 2007. A Comparison of Statistical Significance Tests for Information Retrieval Evaluation. In *CIKM 2007*.

Mihai Surdeanu, David McClosky, Julie Tibshirani, John Bauer, Angel X. Chang, Valentin I. Spitkovsky, Christopher D. Manning. 2010. A Simple Distant Supervision Approach for the TAC-KBP Slot Filling Task. In *TAC 2010 Workshop*.

Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes and Gregory Marton. 2003. Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering. In *SIGIR 2003*.

Ellen M. Voorhees. 1999. The Trec-8 Question Answering Track Report. In *TREC 1999 Workshop*.

Le Zhao and Jamie Callan. A Generative Retrieval Model for Structured Documents. In *CIKM 2008*.