Named Entity Network based on Wikipedia

Sameer Maskey [1*], *Wisam Dakka* [2*]

[1] IBM T.J. Watson Research Center
Yorktown Height, NY, 10598

[2] Google
New York, NY, 10011

smaskey@us.ibm.com, wisam@google.com

Abstract

Named Entities (NEs) play an important role in many natural language and speech processing tasks. A resource that identifies relations between NEs could potentially be very useful. We present such automatically generated knowledge resource from Wikipedia, Named Entity Network (NE-NET), that provides a list of related Named Entities (NEs) and the degree of relation for any given NE. Unlike some manually built knowledge resource, NE-NET has a wide coverage consisting of 1.5 million NEs represented as nodes of a graph with 6.5 million arcs relating them. NE-NET also provides the ranks of the related NEs using a simple ranking function that we propose. In this paper, we present NE-NET and our experiments showing how NE-NET can be used to improve the retrieval of spoken (Broadcast News) and text documents.

Index Terms: Named Entities, Speech Retrieval, Information Extraction, Question Answering

1. Introduction

Identification of NEs has been shown to be useful for text and speech summarization [1, 2], Question Answering [3], and many other tasks. Even though there are many NE detection systems [4, 5] with high accuracy, we do not have many systems that relate NEs. Given a NE, finding related NEs should be useful. For example, having a list of people, locations, and organizations related to Bill Clinton may help in finding the soundbites of Bill Clinton in Broadcast News (BN) even when Bill Clinton may not be explicitly mentioned. Clustering techniques can also benefit from knowing how one NE relates to another by adapting this knowledge into the similarity function.

Several mining approaches using co-occurrence statistics have been proposed to find related NEs [6, 7, 8]. These methods use co-occurrence of two NEs in the web pages or emails to compute if the NEs are related. Due to unstructured nature of the web and the use of co-occurrence statistic that can be unreliable for rarely occurring NEs, relations produced by these methods are noisy. On the other hand [9] manually defines relationship types and uses Wikipedia categories and WordNet to obtain relations such as “Elvis” $born-in$ “1943.” Even though relations produced by this method are more accurate we are limited by the manually defined relationship types. There has also been more recent work on using Wikipedia for various NE related tasks. [10, 11] compute semantic relatedness between NEs using Wikipedia. [11] proposeed path based measures and information content measures to find semantically related NEs. [12] uses Wikipedia to disambiguate NEs. These authors have shown that Wikipedia annotation performed by many humans (thoughy noisy) are reliable enough for many NLP tasks.

Instead of getting humans to label links between NEs or mining the web as some of the work mentioned above, we mine the associations labeled by contributors of Wikipedia pages. This allows us to produce relations that are not noisy and are not limited to manually defined relationship types. Besides finding the related NEs we also provide a ranking function, Inverse Network Frequency (INF), that can be used to rank the related NEs. We show the use of NE-NET and INF in a document retrieval task for GALE [13] queries where we retrieve text and spoken documents from TDT4 corpus. We describe our process of building NE-NET and our ranking function in Section 2 and 3 respectively. We present our experiments showing the use of NE-NET in a document retrieval task in Section 4 and conclude in Section 5.

2. Building NE-NET

We observe that Wikipedia contributors and editors, besides writing and editing Wikipedia pages, annotate each page by identifying important terms in the page and create links between the terms to other existing Wikipedia pages. In many cases, these terms are NEs, which means that contributors have identified important entities and related them with other important NEs in Wikipedia. In other words, if we follow every link between every term in every Wikipedia page, we will have the relationships among every important NE available on Wikipedia. If we encode all of this information in a graph we obtain a network of related NEs, NE-NET, with valuable information about how more than 1.5 million important NEs are related to each other.

Let us consider an example of a Wikipedia page about Orhan Pamuk, winner of 2006 Nobel Prize in literature.[1] The page includes links to more than 90 terms. Many of these terms are NEs that are highly relevant to Orhan Pamuk, such as “Thomas Mann”, “Marcel Proust”, “Leo Tolstoy”, and “Fyodor Dostoevsky.” If we extract these terms and represent them as nodes and link them with arcs whenever there is a link provided by a contributor, we get a graph as shown in Figure 1.

[*] The authors performed this work while they were at Columbia University

[1] http://en.wikipedia.org/wiki/Orhan Pamuk

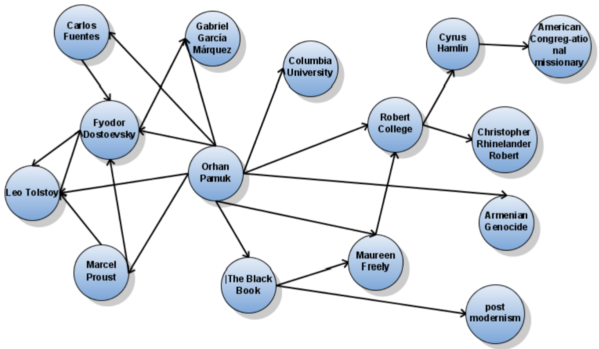

Figure 1: A subgraph of Turkish novelist Orhan Pamuk

We should note all the terms that are linked by contributors in Figure 1 are not NEs (e.g. "post modernism" is linked to "the black book" in Orhan Pamuk's page). In order to filter out non NE terms we use NE classifier particularly built for classifying the terms in Wikipedia [14]. They report very high accuracy for detecting person, organization,location and miscellaneous. We use their classifier in a blackbox approach to classify NEs of our NE-NET.[2].

We should also note that some contributors may identify the same NE with various surface terms. For example, for the node "Orhan Pamuk", contributors use the surface terms "Pamuk," "Orhan," "Orhan Parmuk," and others, including ones with spelling mistakes. One of the biggest advantages we get for using Wikipedia is that such surface variations are identified by the contributors and they point them all to one Wikipedia page. This allows us to search our NE-NET with all possible surface variations including likely spelling mistakes of a given NE.

In order to build our final NE-NET graph we retrieve entire Wikipedia articles and exploit its XML structure to identify the links. There were about 1.5 million nodes with pages which were connected by approximately 6.5 million links. After we build the graph we append each node with the following metadata: (i) **Category** that describe the type of NE (ii) **Surface** that provides all surface variations and (iii) **Abstract**. After various memory optimizations we can load the entire graph in memory which allows us to find related NEs for a very large degree of N.

The first type of metadata field in the node is *Category*. This metadata field carries the classification of each Wikipedia page. It can have one of the following values obtained using [14]'s NE classifier: person, organization, location, misc, and common. The category metadata allows us to build sub-networks of just person names or organization names as well. The second metadata field *Surface* contains the surface forms of the given node. For example, if we look in Wikipedia we will find two surface forms, George Bush and George W Bush, for the page of George Bush. Hence we store both surface forms making our NE-NET robust to spelling variations. The third meta-data *Abstract* provides a few sentences, created by Wikipedia contributors, to describe the node's page.

3. Enriching Nodes with INF

We should note that we can search NE-NET not only for related NEs but also for NEs related with a higher degree of separation than 1. For example, if we search for related NEs of "Orhan Pamuk" with degree of separation of 4 we get hundreds of related NEs. Being able to rank such long list of related NEs can be useful for NLP tasks. We propose a ranking function, Inverse Network Frequency (INF), that ranks the related NEs based on the number of links between the NE and other NEs. INF for a term i is described by the following equation.

$$INF_i = log(\frac{|S|}{\sum e(i,k)}) \qquad (1)$$

where $|S|$ is the size of the graph (the total number of nodes in the graph) and $\sum e(i,k)$ is the sum of edges between the node k and node i where edge originates at k and ends at i and $i \neq k$ and $distance(i,k) = 1$. For our experiments we normalize the INF values by dividing it by $inDegree_i/Outdegree_i$ to compute the ranks of the related NEs. $inDegree_i$ is the number of incoming links and $Outdegree_i$ is the number of outgoing links of a node i respectively.

The INF measure was motivated by a similar measure Inverse Document Frequency (IDF) [15] that is frequently used in IR tasks. IDF is computed by taking a log of total number of documents divided by the number of documents the term occurs in a large corpus. It is frequently scaled using Term Frequency TF of a document to obtain $TFoIDF$ scores that are used to find a relevant set of words to the topic of the document. In a similar fashion we can also obtain $TFoINF$ using INF scores. We should note that in IDF computation, the count of denominator is incremented even though the term may have occurred in the corpus without any relevance to the topic NEs of a given document. In our case, we are able to take account of relevance with NEs of interest because we know NEs are linked by contributors only when they have some relations to each other. $TFoINF$ scores can be computed with the following equation.

$$TFoINF_{i,j} = TF_{ij}.INF_i \qquad (2)$$

where $TF_{ij} = \frac{n_{i,j}}{\sum_k n_{k,j}}$ and the numerator $n_{i,j}$ is the frequency of term in document D_j. The term frequency is normalized by the denominator, the total number of all terms in document D_j.

We show the use of INF based ranking and NE-NET relations in our experiments that we describe in the following section.

4. Experiments and Results

We present our experiments on the use of NE-NET for a task of document retrieval. We performed our experiments using the queries provided for the first and the second year of GALE project. For GALE project, a system has to retrieve documents from a large corpus of text and spoken documents in many languages and produce an answer from the retrieved documents for

[2] More details on NE classifier can be found in [14]

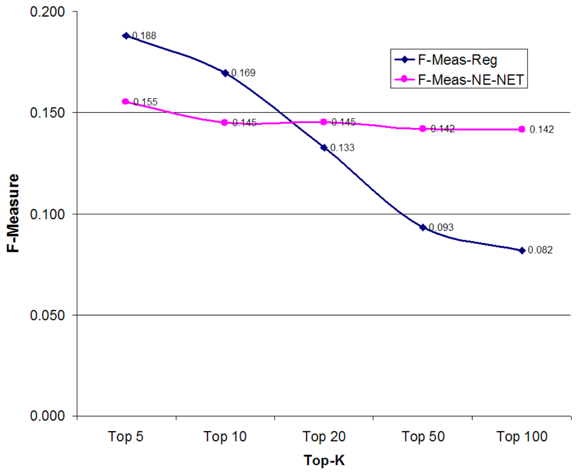

Figure 2: F-Measure curve for retrieval using regular vs. NE-NET queries

the given query. We chose these years of GALE because we had gold standard answers for the document retrieval task.

We first took all the text and broadcast news data available in TDT4 that was provided as a part of GALE corpus and created an Indri[3] index. We then randomly selected 19 GALE queries for which we had gold standard answers. We converted these queries to Indri query format. We used these queries to retrieve relevant documents using Indri search engine on TDT4 index. We compared the retrieved documents with the gold standard answers using information retrieval measures. This setup of our experiment was our baseline.

Next, we wanted to see how we can use NE-NET to improve on our baseline document retrieval system. One simple method that have consistently shown gains in information retrieval community is the technique of query expansion [16]. The query expansion provides gains only if the added lexical items are very relevant to the main query terms. In our case, we can find NEs related to our query NE using NE-NET and add them as additional terms. For example, for the GALE query "Provide information on Ehud Barak" we obtained a regular baseline Indri query that only contained the query-term "Ehud Barak" and the NE-NET query that contained the following list of NEs as relevant terms which were obtained using NE-NET:

- *Ehud Barak, taba summit, dan shomron, bandar bin sultan, yehoshua saguy, limor livnat, amnon lipkin-shahak, ahron bregman, yossi sarid, binyamin ben-eliezer, krav maga, sayeret matkal, tal law, education minister of israel, 1973 israeli raid on lebanon, moshe arens, blue line (lebanon), list of defense ministers of israel, electronic data systems, amir peretz, camp david 2000 summit, meretz-yachad, al-aqsa intifada*

One of the problems with query expansion is that adding terms that are not relevant for query can actually reduce the performance. In our case we may get hundreds of related NEs for a NE term such as "Bill Clinton" which may be linked to many Wikipedia pages. In order to avoid adding all related NEs retrieved from NE-NET we rank them using normalized INF that we described previously, and select only the top NEs. For our NE-NET experiment, we expanded our baseline queries (BQs) to their expanded NE-NET query forms (NQs) using Indri's "#combine" operator where the added terms were obtained from NE-NET. We then used the expanded query with Indri search engine and again retrieved a set of documents. We compared the retrieved documents with the gold standard answers.

We compared the results of these two different experiments using standard precision, recall and f-measure. We tested each query by choosing the Top-K documents of the returned results with $K = 5, 10, 20, 50, 100$. The average precision, recall and f-measure for various K values are shown in Figure 4 and Table 1. We note that regular queries quickly degrade as we increase the value of K, but the f-measure for NE-NET queries degrade very little – even when we choose more top ranked documents. In fact, for top 50 and 100 documents, f-measure for NE-NET queries is higher than for regular queries by 5.97% and 4.86%, respectively.

K	Type	Prec	Recall	F-Measure
Top 5	REG	0.316	0.211	0.188
	NE-NET	0.242	0.188	0.155
Top 10	REG	0.232	0.238	0.169
	NE-NET	0.205	0.206	0.145
Top 20	REG	0.145	0.273	0.133
	NE-NET	0.192	0.217	0.145
Top 50	REG	0.079	0.289	0.093
	NE-NET	0.188	0.217	0.142
Top 100	REG	0.064	0.309	0.082
	NE-NET	0.188	0.217	0.142

Table 1: Precision, Recall and F-measure for various K values.

We see in Table 1 that most of the gain in f-measure is due to better precision. Retrieving more documents with only the query terms quickly degrades for regular queries. For NE-NET queries, our expansion technique provided enough relevant terms to retrieve relevant documents even for higher K values. We see that the precision of NE-NET queries is 12.30% and 10.88% higher for the top 100 and 50 documents in comparison with regular queries. For the same queries we see a degradation of recall. However, the ratio of increase in precision and decrease in recall is high enough that we see significant improvement in f-measure. We observe that when we choose very few documents, such as the top 5 documents, regular queries perform better than NE-NET queries. We expect this behavior because the documents that are ranked in the top 5 or 10 are likely to be very relevant as there will be documents with an exact match to the query terms. Usually the problematic documents are the documents that are ranked lower than the top 10 that do not have exact query terms in them, and these were the documents NE-NET was able to handle better. We should also note that the query set we chose has a mix of different GALE query types. Hence the use of NE-NET is robust enough for us to use in all query types for GALE. The results of our above experiment show that using ranked list of related NEs obtained from our knowledge resource NE-NET can be used to improve text and spoken document retrieval.

One of the reasons NE-NET is useful in retrieving spoken

[3] http://www.lemurproject.org

documents could be because of better handling of ASR errors due to Out of Vocabulary words (OOVs). If the query term we are using is OOV and has been misrecognized by ASR in all of the spoken documents we can still retrieve related documents because ASR may have correctly recognized at least a few of the related terms provided by NE-NET.

We should note that even though we showed improvement in only one type of NLP/Speech task of retrieving text and spoken documents, there are possibly other possible uses of NE-NET. For example, for speech summarization [2] we know that NEs are one of the most important features. If we were to know the relevant NEs, we will be able to weight the sentences further according the distribution of related NEs "possibly" producing better speech summarizer. Similarly, NE-NET can potentially be used for topic based language models. In topic based language models the one of the tasks is to cluster language model data into topics using text similarity methods. Such similarity methods have hard time taking account of relevance in two completely different words such as "George Bush" and "Laura Bush". On the other hand NE-NET can provide the important information that these two words are related to each other and should be considered when clustering sentences containing them. We do not have any formal experiments to show the benefits of NE-NET for some of these "potential" applications; but we do have inclination on reasons for NE-NET to be useful for them.

In particular, we are currently investigating how to introduce NE-NET as a tool to web search of spoken documents to enrich query conceptual representation. For example, consider the query *[black book]*. Using the NE-NET graph, we can enrich this query with the following concepts: *[Orhan Pamuk]* and *[post modernism]*, not just to expand the query but also to perform page filtering and boosting: the ranks of web pages with a reasonable distance from these NE-NET concepts should be boosted while others should be demoted.

5. Conclusion

We presented a knowledge resource Named Entity Network (NE-NET) that can provide related NEs for any given NE. NE-NET has a high precision because it relies on extracted relations between NEs using manually labeled connections in Wikipedia. Moreover, NE-NET has 1.5 million entries which is much larger than other similar resources and it provides a measure INF that can be used to rank related NEs . We demonstrated the effectiveness of NE-NET in retrieving spoken and text documents by expanding queries with related terms found by NE-NET. We saw that as we retrieved more documents NE-NET expanded queries outperformed the regular queries with absolute improvement of 5 to 8%. Even though we presented NE-NET's use in only a retrieval task we hope to show the applications of NE-NET in many other text and speech processing tasks in the future.

6. References

[1] B. Schiffman, A. Nenkova, and K. McKeown, "Experiments in multi-document summarization," in *In HLT 2002.*, 2002.

[2] Sameer Maskey and Julia Hirschberg, "Comparing lexial, acoustic/prosodic, discourse and structural features for speech summarization," in *Proc. of Eurospeech*, 2005.

[3] R. Srihari and W. Li, "Information extraction supported question answering," in *In Proc. of TREC-8*, 1999.

[4] Shumeet Baluja, Vibhu O. Mittal, and Rahul Sukthankar, "Applying machine learning for high-performance named-entity extraction," *Computational Intelligence*, vol. 16, no. 4, pp. 586–596, 2000.

[5] A. Mikheev, M. Moens, and C. Grover, "Named entity recognition without gazetteers," in *In Proc. of EACL*, 1999.

[6] Henry Kautz, Bart Selman, and Mehul Shah, "Referral web: combining social networks and collaborative filtering," *Commun. ACM*, vol. 40, no. 3, pp. 63–65, March 1997.

[7] Peter Mika, "Flink: Semantic web technology for the extraction and analysis of social networks," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 3, no. 2-3, pp. 211–223, October 2005.

[8] A. Culotta, R. Bekkerman, and A. McCallum, "Extracting social networks and contact information from email and the web," 2004.

[9] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum, "Yago: A Core of Semantic Knowledge," in *16th international World Wide Web conference (WWW 2007)*, New York, NY, USA, 2007, ACM Press.

[10] Evgeniy Gabrilovich and Shaul Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis," in *In Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 1606–1611.

[11] Michael Strube and Simone P. Ponzetto, "Wikirelate! computing semantic relatedness using wikipedia," July 2006.

[12] Silviu Cucerzan, "Large-scale named entity disambiguation based on wikipedia data," in *In Proc. 2007 Joint Conference on EMNLP and CNLL*, 2007, pp. 708–716.

[13] GALE, "Gale autonomous language exploitation," 2006.

[14] W. Dakka and S. Cucerzan, "Augmenting wikipedia with named entity tags," in *The Third International Joint Conference on Natural Language Processing (IJCNLP)*, 2008.

[15] Karen Sparck Jones, "IDF Term Weighting and IR research lessons," in *Journal of Documentation*, 2004.

[16] Ellen M. Voorhees, "Query expansion using lexical-semantic relations," in *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, 1994, pp. 61–69, Springer-Verlag New York, Inc.