CREATION OF ENTITY SYNONYMS DICTIONARY AND ITS USAGE FOR QUERY REFORMULATION: A REVIEW

Mamta Kathuria, C.K.Nagpal, Neelam Duhan
YMCA University of Science & Technology

ABSTRACT-The current trend in internet search is to find the similar entities. Keeping information retrieval in mind, there can be many possible way to refer a single entity.The work done in this paper helps to recognize and link different synonyms together and thereby be able to include documents where the entity being sought is included, but the naming is different. The motivation is to build a large dictionary of named entities and their synonyms using various methods used in the literature. This will further help to build a modified search system using query expansion to reformulate the original queries to include synonyms when an entity is detected. This paper also emphaseson the merits and demerits of various methods used in the literature to find entity synonym.

Keywords: **Entity Synonym, Query Reformulation, Entity Dictionary**

I INTRODUCTION

The World Wide Web (WWW) can be viewed as a huge data housewhich keeps information related to almost every domain of knowledge that can be easily accessible whenever needed. It can be seen that user can use different ways to refer to the same entity. For example Barack Obama, Obama, Barack Hussein Obama all refers to the same person who is president of United States. When a query is entered onto the search interface, the search engine first accepts query keywords as an input, search for the specific keywords in its own database and returns the Search Engine Result Pages (SERPs) that contain these keywords. To enhance the power of traditional search engines and to improve the efficiency of the search engine, not only the keyword but synonyms are also considered to find the resulted pages. Synonyms of search queries are quite crucial as they helps in solving the problems that comes in the field of Natural Language Processing (NLP) such as search query expansion, text summarization, text generation etc.For enhancing user search experience, there must be some method which is capable to detect and use semantically similar entities in the query. So there is a need to construct a large dictionary of named entities and various synonyms and utilize them to answer user search query. The entity synonyms have wide heterogeneous variety without following a particular trend making their finding quite a challenging task.

From the literature, it has been observed that the lexical resources can't help us in finding the entity synonyms relating to entities in the common domains such as movies, books, brands, newspaper, restaurants, shopping malls etc. So, there must exist some empirical method based upon the web search and web log which can help to detect entity synonyms.

The subsequent section talks about such empirical methods as used by the various researchers along with their merits and demerits.

II LITERATURE SURVEY

In today's search system, entity synonyms are important ingredients for improving user search results and user satisfaction. It provides improved search relevance and improved user experience. The vertical and web search is having great usefulness of Entity synonym discovery as it helps boost recall, improves precision and enhances user search experience.

To gather entity synonym, manually created knowledge bases such as Freebase[1]and Wikipedia[2] can be used. Freebase consists of semantic knowledge especially for the entity and relation information. In Freebase, entity synonym lists such as aliases) are used for most of the entities. To collect valid entity synonym from Wikipedia, redirect pages and disambiguation pages are used.The Wikipedia was popular lexical resource because of its large size and freely available collection of knowledge. But these resources (such as Freebase and Wikipedia) provide limited coverage and diversity due to their manual creation.

Tao Cheng et.al [3,4] described a method based upon search data and click data to find the set of entity synonym.They have defined two sets. The whole process of entity synonym finding has been divided into three categories:

- Candidate Generation: Wherein the entity synonyms are generated through query strings having some common search data (surrogate pages)and clicked data(opened pages).
- Candidate Selection: Wherein the quality synonyms set are chosen using a threshold value of intersecting page count and intersecting click ratio.
- Noise Cleaning: Wherein both context sensitive and common noise is removed.

They also proposes ClickSim[4] which uses web documents to find semantically similar entities. In order to generate synonyms they use number of documents that are clicked for both entity and its synonyms.

The major achievement of the work is their pioneer effort to find the entity synonyms in automated manner using web query log and search data. However the work suffered from some major issues as listed below:

- Click log sparsity problem that occurs when a query is asked by very limited users and document clicked are very few in number.

- Inability to make a distinction between entities related to different concept and classes e.g. Oracle 10i and Oracle 10i tutorial may be assumed as entity synonyms though they are referring to different concepts.

P. D. Turney(2002)[5] proposed a method to find entity synonym using document similarity. He gives an idea that if two documents are related to each other and one contains the entity reference string, the other contains the candidate synonyms string then the entity reference string and candidate synonyms string are strongly related to each other. The supporting evidence is total number of documents in which both re and ce occurs. The number of documents can be calculated using |aux(re) ∩ aux(ce)|. The function Fdocumentsim(re → ce) is the strength of the relationship between re and ce.

KaushikChakrabarti et.al.[6] proposed a new method which construct a PseudoDoc using query Click log. To construct a PseudoDoc(d) = {y| yεq,s.t. y clicked on d in log L}, the query log can be collected for a time period. The supporting information supp(y) of string y is the set of documents clicked by query y. Formally, supp(y) = {d|y clicked on d}

To check whether a candidate synonym string is covered by queries clicked on a document, we construct the pseudo document for each document. The pseudo document of a document d (referred to as pseudodoc in short) is the set of all tokens from all the queries that clicked on document d.

This method helps to overcome the problems which may occur due to click similarity and document similarity. This method is capable to find even those synonyms that are generally missed by click and document similarity. It may be possible that the document is not clicked for a query, then it can be inferred that q(query) is related to d(document). For example in figure 2 it can be realized that the document d2 is not clicked for the query "data mining". So, the queries that isnot synonymous by document similarity. But pseudo document similarity can determines this. A pseudo document is prepared by collecting all the tokens from all the queries that clicked on document d.

They familiarized the concept of pseudo document similarity to overcome the click log sparsity problem and query context similarity to ensure the definite relationship and belongingness to same concept class. To compute Pseudo document similarity, input query and the candidate synonyms are divided into tokens and a pseudo document (bag of words) is developed by connecting/concatenating the tokens. The pseudo document similarity function ensures the higher recall and consistent precision.

They enhance the concept of entity synonyms by introducing the concept of reflexivity(a string is a synonym of itself), Symmetry(if a string x is a synonym of string y then y is also entity synonym of x), the major requirement for being the entity synonym as imposed by them was the concept of similarity which made it mandatory that in order to qualify as entity synonyms the two strings must belongs to same concept class and must have strong relationship as well. They introduce the concept of auxiliary evidence for defining the quality of entity synonym using the strength of the relation. The auxiliary evidence, aux(s), in their work refers to the number of documents clicked corresponding to a query string s.

To check the relationship between two strings s1 and s2 the pseudo document process involves the generation of a pseudo document d from a document x by combining the tokens of all the queries that clicked on x. The example is given in fig. below.

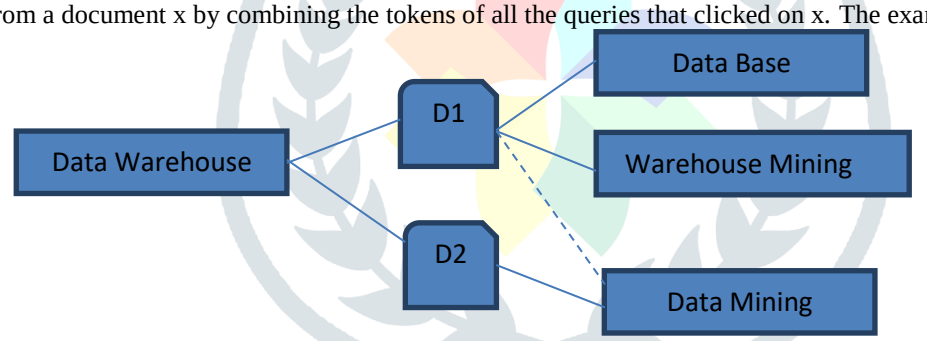

Fig1: Example to add a new edge in click graph using pseudo document

In Fig 1.a new edge can be added into click graph between data mining and d1 using pseudo document even if "data Mining" did not click on d1.

This measure removes the one of problem of click similarity and finds candidate synonyms that belong to same concept class. For example, "Oracle 10i tutorial" could be decided as synonym for "Oracle 10i" because based on supporting evidence they are strongly related to each other. To differentiate between entities that belong to different concept class, the context of entity name is used. Contexts are the word that comes either on the left or right of the query string. Thus, the context set for the query "Oracle 10i" is {download, help, installation guide} and for the query "Oracle 10i tutorial" the context set is {book, ppt, guide}. The QCSimi between re and ce is calculated by considering the set similarity of their contexts. Any measure like Cosine similarity or Jaccard Similarity can be used to find the query context similarity. The supporting evidence here is the set of contexts. In practice, such hard thresholds on each similarity measure are difficult to determine and may be too restricting; a more practical approach is to use the similarity values as features and use a classifier to determine whether re and se are synonyms.

The query context similarity is computed using one way and two similarity functions as follows:

Let s1 and s2 be two candidate strings for entity synonyms. Then one way similarity functions F12 and F21 are computed as follows:

$$F12(s1 \rightarrow s2) = \frac{|aux(s1) \cap aux(s2)|}{|aux(s2)|}$$

$$F21(s2 \rightarrow s1) = \frac{|aux(s1) \cap aux(s2)|}{|aux(s1)|}$$

They further proposed a two way similarity function F(s1,s2) to capture both ways similarity.

$Fqcsim(s1 \rightarrow s2) = Fqcsim(s2 \rightarrow s1) = \frac{|aux(s1) \cap aux(s2)|}{|aux(s1) \cup aux(s2)|}$

They used these two similarity measures by imposing separate threshold for them. These similarity functions were used to compute query context similarity.

They use two measures for effectiveness evaluation of their work:

•Precision: number of true synonyms divided by the total number of synonyms output;

•AvgNumOf Synonym: Average Number of Synonyms Per Entity.

Prabha.P, Dr.P.Sampath(2016)[7] uses feedback session as a database to construct pseudo document. The pseudo document constructed from this method contains all keywords, even keywords from ambiguous query are considered. In feedback session we have both clicked URLs that were clicked by user in a session and unclicked URLs that were not opened by the user in a particular session. It ends with a URL that was clicked lastly in a user session from click-through logs. It contains URL detail along with view detail. So, feedback gives us the more detail than the data obtained from clicked URLs and clustering search result.

Consider a document d $\in$supp(r_e). If all the tokens of c_e contains in d's pseudodoc then r_e is related to c_e. Now the fnctionFpdocumentsim($c_e \rightarrow r_e$) = $|\{d|c_e \in PseuDoc(d), d \in aux(r_e)\}|$. Two way checking of similarity function is performed to ensures symmetry.

It is clear that Pseudo document helps to find even those synonyms which may miss while using click or document similarity. Now, there are some advantages of opting this, these are

- Pseudo document gives a very succinct yet high quality representation of the document.
- The two way checking done in PseudoDocSim helps to improve recall.
- In contrast to DocSim, pseudo document allows us to focus on the essential parts of a document, rather than the complete content.Mining over pseudo documents yields higher precision, as compared to the document similarity approach.
- Since pseudo documents are much shorter than real documents, it is much more efficient to compute as well.
- When compared with ClickSim, PseudoDocSim outperforms better by outputting more number of synonyms per entity with high accuracy.
- PseudoDocSimhelps to removesparsity problem of ClickSimand output more number of synonyms.
- The improvement in recall is due to the fact that search engines'ability to direct a query to a document (by query alternation, spell checking or partial match) although the query does not appear exactly in the document.

Till now the techniques discussed are applicable only to offline and structured data and not to the dynamic and unstructured WWW. The algorithm discussed by Srikantiah K.C et.al.[8] solves the problems related to keyword based approaches..

The problem related to keyword based approach are as follows:

1. The irrelevant URLs may be returned by the search engine, even if the given keyword is present in them.
2. Some relevant URLs may not be returned by the search engine, because the synonym of the keyword is present in them, not the original keyword.

The ASWAT algorithm is capable of working in a dynamic, online environment and it is not domain-specific

The algorithm proposed by Srikantiah K.C et.al.[8] provides a new way to solve the problems related to keyword based approach. The anchor text used by them is treated as a clickabletext in a hyperlink and is significant to the page a user is looking for, rather than generic text.

The algorithm used Search engine result pages to extract the anchor text present in the URL and generates a ranked list of candidate synonyms for query keyword using co-occurrence frequency and page count based measures. To prove the relevance of the generated synonym the URLs described by the synonyms gets compared with the URLs retrieved from the original query keyword. This new technique is domain-independent and scalable.

Automatic Discovery of Synonyms from the Web using Inbound Anchor Text (ASWAT) is proposed that generates a ranked list of synonyms. In order to generate the candidates, the URLs obtained from the SERPs are compared by querying both the original keyword and its subsequent results. The main insight of using this new method is to extract the inbound anchor text(i.e candidate synonym) whenever there is a match occurs between the initial and subsequent URLs.

Candidate Synonyms: Synonyms of a keyword A are defined as the anchor texts that exactly have the same URLs linked to them as that of A.

Inbound Anchor Texts: refers to a set of anchor texts that are pointing to the same URLs that are relevant to the search keyword.

ASWAT algorithm finds the candidate synonyms using inbound anchor text by entering the query keyword A onto the search engine S to get the SERPs. Then the all the URLs from the SERPs are collected to form a set of parent URLs i.e. ParUrl.

ParUrl=$\{u_i | u_i$ is a URL $\in$ SERP and $\forall i, 1 \leq i \leq n\}$.

Where, n is the total number of URLs present in all SERPs for query A. Then, for each parent URL $u_i \in$ParUrl, the set of pages that are linked to u_i are retrieved. In other words u_i is send as a query to the search engine to generate the SERPs corresponding to u_i and the subsequent URLs contained in them are collected to form a set of sub parent URLs SPU, i.e.,

Now in order to rank these candidate synonyms, Co-occurrence Frequency (CF) is used that counts the number of distinct URLs between keyword A and its candidate synonym. The higher value of CF determines the more relevancy of the candidate synonym. So, the decreasing order of CF is considered to rank the candidate synonyms. Hence the candidate with the highest CF value will be ranked in the first position and so on.

After the effective execution of this algorithm, a ranked list of candidate synonyms is provided. This algorithm can be executed on the dynamic web and unstructured data. The main advantage of this technique is scalability and solves the problem related to polysemeous words. It provides the accurate list of synonyms.

As discussed above, user can extract entity synonyms by analysing different kinds of information in the query log, such as query click-through data , query context, pseudo-document for web page and their combinations [9]. However, existing query log-based methods encounter two major limitations as follows:

a) Ambiguity of entity string name

Name ambiguity arises when an entity string name is given as an input. For eg. Ram Mehra may refers to a famous T.V actor or Mehra& Sons manufacturing company. So Synonyms generated from click logs and context based method are ambiguous due to click statistics and noisy context.

b) Ambiguity of Synonym String

This problem arises due to the weak click statistics between the web page and target entity. The existing work uses web queries to generate synonyms but it does not take into consideration the sub-queries as a whole. Therefore there is a need to recognize such sub-queries for identifying true synonyms.

Xiang Ren et.al[10] proposes a new method that focuses on the structured view of an entity instead of abstract view(i.e. string name) The main focus of this method is to explore sub-queries, to explore tailed synonyms and tailed web pages for gathering more synonyms.

The structured attribute helps to crisply define an entity and therefore reduce ambiguity. The effort in this paper discover high quality entity synonym with good coverage. They focus on two attributes i.e. entity source web pages and already existing synonym from knowledge base.

They proposed a model that take structured entity e=(re,Ue,Ce) where re stands for reference name of the entity e, Ue are the set of web pages and Ce are set of existing synonyms.

Xiang Renet.al[10] proposes a heterogeneous graph-based framework, called StrucSyn, to discover entity synonyms, using three types of objects, i.e., synonym candidates, keywords and web pages. This paper discovers synonyms using graph and reveal optimal solution

The existing method like knowledge bases uses various spelling variants as candidate synonyms. The proposed method focuses on finding not only these spelling variants but discovers more semantic synonyms.

This method works by focusing on two types of structured attributes, i.e., entity source web pages and existing synonyms. These attributes are available in entity knowledge bases and they are domain independent.Because the source web pages are less ambiguous than entity name therefore they generates better quality of entity synonyms. The aim of this work is to gather entity synonyms from click-through data of web search queries

Specifically, user click-through data L is a set of tuples l = (q,p,n), where q is query, p is web page and n is the number of times users have clicked on web page p after issuing query q. With the definitions of structured entity, entity synonym and query log, the formal data for Structured Entity Synonym Discovery is as follows:

Input is a click-through data L and structured entity e= (re,Ue,Ce) and output is a list of candidate synonyms and entity synonym score for each candidate synonym. The highest value of entity synonym score is used to find the best entity synonym for a given input string.

The idea of this paper is to explore not only queries but also to focus on sub-queries for the candidate generation process. The existing work focus only on the co-click queries and ignores sub-query synonyms that are the silent features for harvesting more candidate synonyms. The co-click queries are used because true entity synonyms always appear as web queries themselves. The main insight here is to extract entity mention from co-click queries.

Co-click queries are those that share many web pages with the input string entered as a query. Then n-grams are extracted from co-click queries. N-grams are word sequences of different length n with a condition that it does not start or end with number or stop words.

In order to produce candidate two information sources are used. First source is the web pages that are clicked by candidate support query and second is the keyword that appears along with candidates in the support queries. The reason for picking web pages are they offers descriptive information about web pages and keywords(equally important) acts as a contextual information about the entity. The importance of a web page can be judged by considering the click count on the web page for a given entity name. The keywords that come along both sides of an entity are equally important. Thus keywords and web pages provide more focused entity synonyms. To check the whether a web page is a true entity page for any entity, page score function is used. Also likeliness of entity synonym can be checked by checking frequency of co-occurrence with the context keyword.

Now to reach the tailed web pages, the query must be reformulated by expanding entity name with entity contexts. So, the seed query must be extended using entity contexts in the candidate generation process. The entity contexts are identified from the web pages that are clicked for a given entity by picking the queries from the query log. The

In practice, we observed that users tend to reformulate queries to reach these tailed web pages, by augmenting entity name with entity contexts (see"Delaware chicken"). It is thus natural to extend our seed query, i.e., entity name, into a set of seed queries for candidate generation, so that more tailed web pages can be included. Specifically, given a structured entity e = (re,Ue,Ce), we first extract entity contexts for e, by exploiting queries which click on source web pages Ue, i.e., Q0 =Su∈Ue NQ(u). For queries q ∈ Q0 containing entity name re as substring, we extract unigrams from q, excluding re, stopwords, and numbers. By doing so, a set of high quality entity contexts, We, can be collected, and will be further used to search seed queries in query log L. In our implementation, we search over L to find queries which contain re, and at least one entity context from We, leading to a set of seed queries, Qe. By replacing re with Qe in candidate generation step, we can obtain a more comprehensive set of web pages, i.e.,Sq∈Qe NU(q), compared to NU(e). Following the same steps in the rest of candidate generation, an extended graph can be constructed to include more tailed web pages. The table 1.shows the different methods used to identify the entity synonyms along with merits and demerits

Table 1 Entity semantic similarity method with advantages and problems

Techniques for Entity Similarity	Advantages and input used	Problems
Freebase	• Use Alaises • Resultant entities are only acronyms and nicknames	• They generally use spelling variants as synonyms • Limited coverage • Limited Diversity • Few or no synonym for less popular entities
Wikipedia	• Based on redirect pages and disambiguation pages • Wikipedia is that it is made up of a large amounts of semi-structured data and we think that it would therefore be a good candidate for data mining.	• Limited coverage • Limited Diversity • Few or no synonym for less popular entities • Manual effort based approaches, such as Wikipedia redirect or disambiguation pages, can be of high quality, but are rather limited to only very popular entries
Click Log Similarity	• Automated approach • Use query log • This could result in a much smaller, but highly relevant collection of named entities and synonyms	• Click Log Sparsity • Inability to entities of different concept class • Problem related to precision and recall • Ambiguity of entity string name • Ambiguity of synonym string • This approach finds some related entities but often they were only related, and not proper synonyms.
Pseudo Document Similarity	• Higher recall without drop in precision • Resolve the problem of Clicklogsparsity • It focus on essential part of the document rather than complete document • Computation easy • Filter the bad quality entity synonyms	• Ambiguity of entity string name • Ambiguity of synonym string
Query Context Similarity	• Find synonyms that belongs to same concept class • Filter the bad quality entity synonyms	• Technique is applicable only to offline and structured data and not to the dynamic and unstructured WWW.
Document Similarity	• Use clustered documents to find similarity	• It suffers from noise in document content • Symmetry property does not hold due to one way checking of similarity function.
Anchor Inbound Text similarity	• Use clickable text as hyperlink • Provide Ranked list of candidate synonyms • This technique can be implemented in the unstructured and dynamic web • Resolve the polysemy problem to large extent • This technique is scalable and domain independent	• ASWAT algorithm is capable of working in a dynamic, online environment and it is not domain-specific. • To extract anchor text
Heterogeneous Graph Based Approach	• Considered not only string name but important structured attributes • It also explore sub queries, tailed synonyms and tailed web pages • closed-form optimal solution for outputting entity synonym scores	• The technique uses entity source web pages and existing synonyms. They are generally available in entity knowledge bases and are domain independent.

CONCLUSION

This paper helps for creating entity dictionary from different approaches used in the past. The work can be used for the purpose of query expansion because popular entities had very large amount of synonyms with very small variations. This will help to improve precision and boost recall.

REFERENCES

[1] K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor," Freebase:a collaboratively created graph database for structuring human knowledge", *In SIGMOD*, 2008.

[2] Strube, M. and Ponzetto, S.P. (2006). Wikirelate! Computing Semantic Relatedness Using Wikipedia.Proc. 21st Nat'l Conf. Artificial Intelligence.

[3] T. Cheng, H. W. Lauw, and S. Paparizos, "Fuzzy matching of web queries to structured data", *In ICDE*, pages 713–716, 2010.

[4] T. Cheng, H. W. Lauw, and S. Paparizos, "Entity synonyms for structured web search", *IEEE Trans. Knowl. Data Eng.*, 24(10):1862–1875, 2012.

[5] P. D. Turney, "Mining the web for synonyms: Pmi-ir versus lsa on toefl", *In Proceedings of the 12th European Conference on Machine Learning, EMCL '01*, pages 491–502, London, UK, UK, 2001. Springer-Verlag.

[6] K. Chakrabarti, S. Chaudhuri, T. Cheng, and D. Xin, "A framework for robust discovery of entity synonyms", *In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '12*, pages 1384–1392, New York, NY, USA, 2012. ACM.

[7] Prabha.P, Dr.P.Sampath, "Optimum Semantic Similarity Search for Pseudo Keyword Terms for Large Scale Scientific Data Collection",International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN: 0976-1353 Volume 21 Issue 2 – APRIL 2016.

[8] Srikantiah K. C., Roopa M. S., Krishna Kumar N., Tejaswi V., Venugopal K. R. and Patnaik L. M., "Automatic Discovery of Synonyms from the Web based on Inbound Anchor Text", *ICDMW 2013*, pp. 130–139. © Elsevier Publications 2013.

[9] Yanen Li , Bo-June (Paul) Hsu , ChengXiangZhai , Kuansan Wang, "Mining Entity Attribute Synonyms via Compact Clustering", CIKM'13, San Francisco, USA , Oct. 27-Nov. 1, 2013, ACM

[10] Xiang Ren, Tao Cheng, "Synonym Discovery for Structured Entities on Heterogeneous Graphs", *International World Wide Web Conference Committee (IW3C2), WWW 2015 Companion*, May 18–22, 2015, Florence, Italy.ACM 978-1-4503-3473-0/15/05.

[11] S. Chaudhuri, V. Ganti, and D. Xin, "Exploiting web search to generate synonyms for entities", *In Proceedings of the 18th international conference on World wide web, WWW '09*, pages 151–160, New York, NY, USA, 2009. ACM.

[12] S. Chaudhuri, V. Ganti, and D. Xin, "Mining document collections to facilitate accurate approximate entity matching",*Proc. VLDB Endow.*, 2(1):395–406, Aug. 2009.

[13] P. Pantel, E. Crestan, A. Borkovsky, A.-M.Popescu, and V. Vyas, "Web-scale distributional similarity and entity set expansion",*In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing:* Volume 2 - Volume 2, EMNLP '09, pages 938–947, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[14] Rodriguez, M.A. and Egenhofer, M.J. 2003, "Determining Semantic Similarity Among Entity Classes from Different Ontologies", *IEEETramsactions on Knowledge and Data Engineering*, 15(2):442{456, March/April.

[15] O. Etzioni, M. Cafarella, D. Downey, A.-M.Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, "Unsupervised named-entity extraction from the web: An experimental study", *Artificial Intelligence*, 165(1):91–134, 2005.

[16] Jun Zhu, ZaiqingNie, Xiaojiang Liu, Bo Zhang, Ji-Rong Wen, "StatSnowball: A Statistical Approach to Extracting Entity Relationships", *In Proceedings of the 18th international conference on World Wide Web (WWW)*, 2009, pp. 101—110. [DOI: 10.1145/1526709.1526724]