The 2015 Conference on Computational Linguistics and Speech Processing
ROCLING 2015, pp. 248-262

Learning Knowledge from User Search

Yen-Kuan Lee, Kun-Ta Chuang
Dept. of Computer Science and Information Engineering,
National Cheng Kung University
E-mail:yklee@netdb.csie.ncku.edu.tw,
ktchuang@mail.ncku.edu.tw

Abstract

In this paper, we introduce the concept of a novel application, called Knowledge Learning from User Search, aiming at identifying timely new knowledge triples from user search log. In the literature, the need of knowledge enrichment has been recognized as the key to the success of knowledge-based search. However, previous work of automatic knowledge extraction, such as Google Knowledge Vault, attempt to identify the unannotated knowledge triples from the full web-scale content in the offline execution. In our study, we show that most people demand a specific knowledge, such as the marriage between Brad Pitt and Angelina Jolie, soon after the information is announced. Moreover, the number of queries of such knowledge dramatically declines after a few days, meaning that the most people cannot obtain the precise knowledge from the execution of the offline knowledge enrichment. To remedy this, we propose the SCKE framework to extract new knowledge triples which can be executed in the online scenario. We model the 'Query-Click Page' bipartite graph to extract the query correlation and to identify cohesive pairwise entities, finally statistically identifying the confident relation between entities. Our experimental studies show that new triples can also be identified in the very beginning after the event happens, enabling the capability to provide the up-to-date knowledge summary for most user queries.

Introduction

The technology of Knowledge Bases, abbreviated as KB, is recently highlighted by

Internet giants such as Google, Yahoo and so on. For example, the Knowledge Graph project 1 , announced by Google at 2012, attempts to integrate the semantic information from KB into the search engine, enabling the capability of Question-Answering for specific queries. Currently, when we issue 'Avatar Director', the exact answer 'James Cameron' will be revealed as the conspicuous block in the search result. As the evolution of the interactive interface moving toward small screens (such as smart phones or wearable devices), the search content with lots of relevant URLs is no longer considered as an effective manner. It is believed that the QA-based search engine is the key ingredient of the next-generation information technology.

However, the public large-scale knowledge bases, such as crowd-based Dbpedia [1], Freebase [3], NELL [4], and YAGO [26], have been reported to encounter the progressively slow growth of content [27]. The bottleneck implies that the human editing is no longer the effective manner for knowledge maintenance. Since the size of knowledge in current KBs still deviates far from completion, Google recently devotes to develop a systematic solution, called Knowledge Vault [9] (called KV for short), for the purpose of knowledge enrichment. As the full scan of Google-indexed pages, the KV framework is able to annotate new relation between two known entities such as people or movies. The design of KV is based on the fact that some relations, such as nationality of people, should be innate so that such relations are likely to be discovered after exhaustive search on the whole web. After the content match of entities, corresponding to Zappa and Rose in this case, the 'parent' relation can be further identified by the technology of text understanding. As their evaluation, the knowledge size is finally enlarged by 100 times as compared to the current knowledge base. Knowledge Vault highlights the necessity of knowledge enrichment. Unfortunately, the strategy of exhaustive scan of the whole web is not scalable to capture the daily updated knowledge, which is believed to be of interest to most

[1] http://googleblog.blogspot.co.uk/2012/05/introducing-knowledge-graph-thingsnot.html

people.

Motivated by this, we explore in this paper a novel problem, called Knowledge Evolution, to discover the daily variant knowledge. The problem of Knowledge Evolution specifically consider two kinds of knowledge triples which cannot be properly identified in previous work: (1) triples with timely updated relations; (2) triples with a newly identified entity and relation.

However, the identification of knowledge in the evolutional sense, although very promising to search engine, still poses a significant challenge to current methodologies. The first challenge results from the noise in the source content. Furthermore, the knowledge in web pages may only be temporally true. For example, most web contents state the ’partner’ relation between Pitt and Jolie, but however, only a few news-based pages start to describe their ’spouse’ relationship after 2014, leading to temporally inconsistent knowledge.

Essentially, systematic knowledge discovery, such as Google KV, pursues the knowledge enrichment, and may finally overcome the aforementioned challenges by sophisticatedly re-designing the framework. Unfortunately, as we discussed, the computational overhead of KV cannot timely identify the presence of new triple at the moment close to the time that the event was first revealed.

The nature of capricious user interest is critical to the system design, but also inspires us to develop the framework of Search Correlation Knowledge Evolution, abbreviated as SCKE, to incorporate user intention into the identification of new knowledge triples. Generally, human issue queries of interest, and check news, blogs, twitter or facebook page for the desired answer. Their search intention may come from word-of-mouth communication, knowing some information about entities. For example, fans of Angelina Jolie once hear her marriage information. They may issue ”Angelina Jolie” or ”Angelina Jolie marriage”, searching the content in news. On the other hand, fans of Brad Pitt may issue ”Brad Pitt”, and also check the identical page for the detailed information.

In this paper, we explore the keyword temporal correlation phenomenon from the daily query log of search engine, which contains the user query and the URL list of clicking. We model the ’Query-Click Page’ bipartite graph to extract the query correlation and to identify cohesive pairwise entities, which correspond to the pair of entities having new knowledge with high probability. The second step of relation identification will statistically identify the confident relation between entities, constructing the new knowledge triples. The SCKE framework is designed with high efficiency since the search space of knowledge identification from search log is significantly pruned. In our internal use, the flow of SCKE can be finished within two hours in the hadoop farm, daily extracting more than 500 new knowledge triples. The new triples can also be identified in the very beginning after the event happens, enabling the capability to provide the up-to-date knowledge summary for most following queries.

The remainder of this paper is organized as follows. Section 2 gives related works. In Section 3, the design of the SCKE model and algorithms are discussed. The experimental results are shown in Section 4. Finally, this paper concludes with Section 5.

Related Work

Knowledge bases has been comprehensively developed for a while in the literature, including the public Dbpedia [1], Freebase [3], NELL [4], and YAGO [26]. These public KBs, most collected from human editing knowledge, are usually utilized as the basis to construct the specific knowledge representation. In this section, we discuss relevant methodologies which aim to automatically identify the knowledge structure. Specifically, knowledge bases can be modeled as a graph, in which the node denotes as an entity and the edge is used to represent the relation between two entities. In the literature, a relevant research topic, called link predication, is to identify whether a link, or relation, exists between two given nodes [18][20][25]. The works of link prediction utilize the network traversal manner to predict the confident link that should exist in the network. Note that the link prediction problem is orthogonal to our work. For the knowledge evolution problem, the necessary information of new triples, which usually comes from external sources, is not embedded in the network. So that the solution of link prediction is difficult to be extended to detect updated knowledge as our need.

In [28], Jun Zhu et al. proposed a method to predict whether it exist a relationship between two entities by using the discriminative Markov logic network [8]. Unlike the traditional relation extraction methods, the work is used to predict whether a token is a relation keyword instead of pre-specifying relations between entities. The input of their system is an initial model formed by the input webpages, and a small set of augment seeds is composed of two entities and zero or one relational keyword. By applying an on-line learning model to compute the probability of two entities with a relation, denoted by $p(R(e_i,e_j)|O)$, where e_i and e_j represent two different entities. R represents a relation and O denotes the observations which can be predicted. In such a way that they can decide the relation between two entities in a maximum likelihood estimator. They finally developed a working entity relation search engine named Renlifang. However, as reported in [9], the work with the method of Open IE inherently faces the issue of data fusion and cannot deal with the timely data source, and thus fails to accommodate to the issue of knowledge evolution.

The other category of algorithms utilizes the random walk approach to traverse the graph and retrieve the knowledge [2][5][16]. Among them, Ni Lao et al. [16] proposed an association rule mining on knowledge base to identify associative rules between entities. Afterward, they apply a random walk strategy and the Path Ranking Algorithm[15] to update the missing triples. Similar to the methods of link prediction, the solution of random walk needs the network traversal and cannot refer to outside knowledge for updated information.

The knowledge vault [9] was proposed at 2014 for enriching annotated triples in the knowledge base. The motivation of the knowledge vault is that there are many knowledge which are unannotated when users wrote the content of knowledge, such as wikipedia. For example, there are 71% of people in Freebase have unknown place of birth, and 75% have unknown nationality. Knowledge Vault is a web-scale probabilistic knowledge base which combines the existing knowledge repositories with the knowledge extracted from web content. The method of the knowledge vault is building a enormous E x P x E three dimensional binary matrix G, which E represents the number of entities and P represents the number of relations (or said as predicates) coming from the fixed ontology such as YAGO. In KV, the value of G(S,P,O) is 1 when a triple S,P,O exists. In contrast, the value of G(S,P,O) is 0.

Knowledge vault employs supervised machine learning methods to fit probabilistic binary classifiers which can compute the probability of a triple and the correctness of a triple. The feature of the classifier is extracted from the whole web. There are four types of the webpages: Text documents, HTML trees, HTML tables, Human Annotated pages. Knowledge vault implements different ways to extract the features from each type of the webpages. After running the classification model, the system retrieves the triples (s,p,o) which the probability of (s,p,o) is higher than the threshold so that the triples can be regarded as the candidate triples. The candidate triples are then inserted to the existing knowledge base. It is possible that some candidate triples conflict with the prior triples in the knowledge base, and so that the system computes the probability of the candidate triples, based on agreement between different extractors and priors to decide whether adding the candidate triples to the knowledge base.

However, as they also mentioned in their work, some critical limitations still needs further justification:

How to choose the correlated sources from the web to extract the daily knowledge. Some facts are not always true but changeable, such as the team of an athlete.

The system builds the fixed size three dimensional binary matrix for the classifiers, so it is a challenge to add new entities and relations in this method. Motivated by resolving these limitation, we propose in this paper the knowledge evolution system, to complement the current offline-based automatic methodology for knowledge enrichment.

The SCKE Framework and Algorithms

The Search Correlation-based Knowledge Evolution

To achieve the goal of knowledge evolution, we propose the system which is based on user logs. The system can be mainly divided into two parts. First, we want to find out the possible cohesive pairwise-entities which indicate that there is some relationship between two entities. We then name this step as Cohesive Pairwise-entities Generation. After finding the possible Pairwise-entities, we need to figure out what kind of the relationship is between two entities by analyzing the content of the related web-pages. The second step is named as Relation Identification. The following algorithms in this paper are under the framework of Map-Reduce, so we list the procedure Mapper and the procedure Reducer in each algorithms.

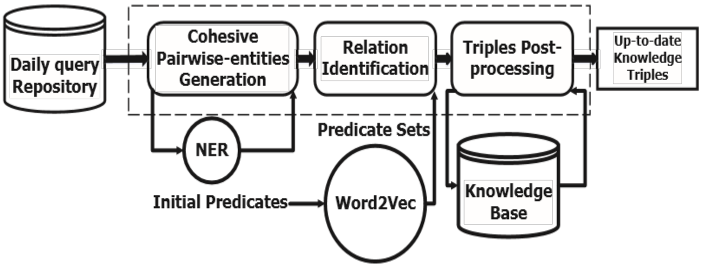


Figure 1: System flow chart of the SCKE framework

Cohesive Pairwise-entities Generation

A cohesive pairwise-entity is composed by two entities if we believe that there is relation between these two entities. When people try to search for something, it is likely that they would make different queries to search for the same information. That is to say, for users who gave queries, if most of the links they clicked are identical, we can assume that there is a relationship between two queries. After processing the query to the entity type, we can consider that these two entities can form an cohesive pairwise-entity. And, just like the link prediction problem, we tend to generate as many as possible cohesive pairwise-entities in step one.

Predicate Set Generation

In the knowledge base graph, an edge between two nodes represent the relation of these two entities. We called it predicate. Predicate is a label of the edge in the knowledge base. There are many kinds of predicates which are predefined by some organization, such as Yago, Dbpedia. There are hundred of predicates in Freebase now, such as ”spouse”, ”cast”, ”parent”, ”partner” and so on.

After indexing the web-pages, we start to work on the important part of this process, which is to extract the predicate from the content of the webpages. It is intuitive that we might find out that there are lots of alias to represent the same predicates. For example, people can describe the relation between Brad Pitt and Angelina Jolie in many expression, like ”spouse”, ”wife”, ”husband”, ”mate”... and so on. To solve the aforementioned problem, we have to enrich the existing predicate word to a predicate set which is composed of many synonyms. We apply the technology named Word2Vec to deal with the predicate expansion.

Word2Vec [6][24][23][12] provides an efficient implementation for the continuous bag-of-words and skip-gram architectures. It uses vector to compute the similarity of words. It is trained by millions of articles. For each word in the article, we compute the cosine distance between one word and the other words. The return score would be the similarity between two words.

For each words in an article, skip-gram means a word will span others words before and after this word in the distance n like the concept of projection. Now, we have two words in the articles, and we want to know if these two words are similar. If the text before or after these two words are very similar in the training articles, the projection of skip-gram of these two words should also similar. Then we can get the high score between these two words. After training the model, we can send a word as the input, and the system will return K words which have the highest score with the input word.

In our system, we use the Word2Vec library which is provided by Apache Spark. The model is trained by using approximately 100 billion words in parallel. And, we use the predicates which are predefined by Freebase as the input. For each input predicate, we retain top 30 highest score words, and choose the suitable words for predicate expansion. Finally, we collect the input predicate and others words expanded by Word2Vec as a set. Then, we named the result as ”Predicate Set”.

Relation Identification

We want to find the most suitable predicate set to represent the relation between entity A and entity B.

In the previous section, we introduced the process of distinguishing the subject and the object in a cohesive pairwise-entity. Suppose that entity A is the subject, and entity B is the object. In an article, it is intuitive that the subject is mentioned more times than the object because the article might also refer other relations between the subject and other entities. So, it’s difficult to determine the relation through observing the subject. Instead, we observe the object to find its relation to subject.

We consider the predicate set which contains w is likely to be the relation between the subject and the object. Also, the relation is considered to be the main message that the article wants to convey. It is said that the closer distance between w and the object, the more likely w is the precise relation between the subject and the object.

Triples Post-processing

By applying the current knowledge base, we can split the output triples into two types. The first type of the output triple is that the triples are already existed in the knowledge base and there are some recently events which lead user to query them. We called it ”the reconfirm triple”. For instance, we retrieve many reconfirmed triples which are the spouse relationship between two idols. The reason of the reconfirmed triples appearing in the hot query again is that there might be some new events related with these two idol like they have a new baby or else.

The second type of the output triple which does not exist in the currently knowledge base, we named it ”the updated triple”. The Updated triple means the relationship detected by our system is distinct or not existing in the knowledge base. However, there are some situations should be treated differently. Based on the different situations, We categorize the update triples into the following cases. (1) The cohesive pairwise-entity is in the knowledge base, but the relation is different from the output of our system. We then update the relation to the current knowledge base.

The cohesive pairwise-entity is in the knowledge base, but there is no relation between them. We will first check if the relation is suitable for those two entities. After checking the correctness, we update the relation we found into the knowledge base.

If there is one of the entity not in the knowledge base, we will apply the NER system to recognize if it is a new entity. If so, the entity will be added ,then, combined with the other entity and their relation to form a new triple.

The applications of the two type triples are different. The reconfirm triples represent the trending of the cohesive pairwise-entities which are popular with many users, and it can be applied to the search engine. For some famous search engine, it will return the ”knowledge graph” when user query an entity. The knowledge graph regarded the entity as the subject entity, and show some of the predicates and the correspond object entities from the knowledge base. Those search engine websites would apply some algorithms to determine which attributes or predicates related to other objects should be displayed on the knowledge graph.

By those triples which need to be reconfirmed, we can know what people want to know recently. That is, reconfirmed triple means it is popular. This message could be the reference of whether this triple should be displayed in the knowledge graph. Nowadays, if you use the mainstream search engines to query the subject entity and its predicate at the same time, the search engine would show the knowledge graph of the subject entity. For example, if user queries ”Director of Avatar”, it will return the knowledge graph of ”James Cameron”. To complete the above task, the search engine uses the triggering list to index the map of the ”subject predicate” and the ”object” rather than traversing the knowledge base in real-time query to find the precise subject entity. Because it would take a long time for browsing through all of the knowledge base to find the corresponding subject. Though looking up the result from the triggering list is more efficient than directly finding the target entity relation, it need lots of time to re-compute the relation. So, the information in knowledge base is lack of flexibility and the data might not be the latest information. The website spend lots of time to compute those data to find all the relation between those entities. But, the truth is that people care little things. The website maintains a thoroughly knowledge base but we know that people often query the same popular entities and they are just a small part of the knowledge base. To make the query more efficient, we use the reconfirmed triples to construct a small world knowledge base graph. With that knowledge base graph, people can get the result in a short time by traversing small graph and it also provide multiple predicate searching that it is not possible to do so in the aforementioned triggering list.

Experimental Results

We signed a contract with Yahoo! and use the user query log for out experiment. After the execution of named entity recognition, we transform the user query terms to the correspond entities. We count the type of the entities and show the statistic result in Table 1.

Type of entities	count
People	1,993,606
Movie	592,351
tv-show	141,250
event	25,550
other	198,843

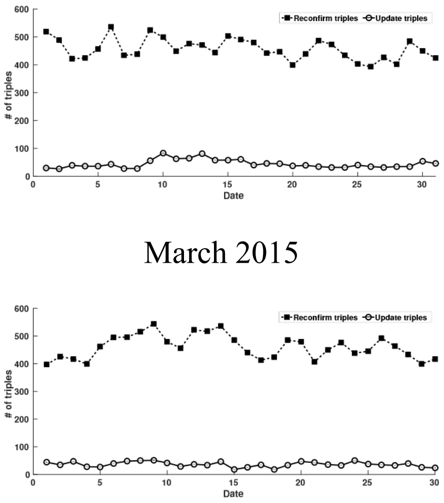
Table 1: Entity types distributed in the March 2015 user query log

In Table 1, type ”People” is accounted for 68% except the unknown type entities. We know that the entity type ”People” is numerous search in the query log. The type ”People”, ”Movie” and ”tv-show” are totally accounted for 92%, which are highly association with the type ”People” to compose to a triple. In the other words, the type ”People” is very frequent and must be an important source to enrich from the knowledge base. For these reasons, we will control one of the entity type in one triple must be ”People” in our experiment.

We generate 10 predicate sets which is used to describe the relation between the entity type of ”People” and ”People”, ”Movie” or ”tv-show”. In the 10 predicate sets, it contains the generally existing predicates are predefined in the knowledge base including ”spouse”, ”cast”, ”parent and child”, ”brother and sister”, ”family”, ”friend” and ”partner” and the user-interested predicates like the ”affair”, ”break up” and ”bad relation” which are found by our observation in the query log.

After the execution of our system, the system would return the score of each predicate sets of each pairwise entities. In our experiment, we set the parameter D = 10 and n = 5 of the function ”RID”. To reduce the error rate, we give a strict condition that if the predicate score is accounted for more than 80% of the total score, we return this predicate set as the relation of the pairwise-entities.

The following figure shows the distribution of the reconfirmed triples and the updated triples. The horizontal axis represents the the date and the vertical axis represents the number of the triples for each types. As an output of the system execution, we get hundreds of the triples which are qualified by the constrains we set everyday. These output triples represent a feature



March 2015

April 2015

Figure 2: Reconfirm and update triples

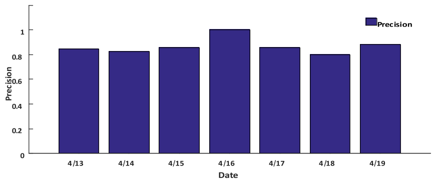


Figure 3: One week precision of the update triples

After checking not only by current knowledge base but also by the strict condition of the post-processing of 4, the precision of the re-confirm triples are almost 100% in our observation. We compute the precision of the update triples in one week showed in the previous figure. The precision of each days are all more than 0.8. Conclusions

Conclusions

In this paper, we first explore a novel problem, called Knowledge Evolution, to identify timely knowledge triples. While previous work all focus to extract knowledge triples by matching the web-scale content, we first attempt to apply the user search intention into the knowledge extraction. The SCKE framework is devised to figure out the clue of new knowledge triple from user search log. Our experimental studies show that new triples can be identified in the very beginning after the event happens, enabling the capability to provide the up-to-date knowledge summary.

Acknowledgement

This paper was supported in part by Ministry of Science and Technology, R.O.C., under Contract 104-2221-E-006-050. In addition, the authors especially thank the Taiwan Ministry of Economic Affairs and Institute for Information Industry for financially supporting this research : “Plan title : Fundamental Industrial Technology Development Program (3/4)”.

References

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. G. Ives. Dbpedia: A nucleus for a web of open data. In *SEMWEB*, 2007.

[2] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WSDM*, 2011.

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

[4] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, 2010.

[5] H. Chen, W. Ku, H. Wang, L. Tang, and M. Sun. Linkprobe: Probabilistic inference on large-scale social networks. In *IEEE ICDE*.

[6] A. Demski, V. Ustun, P. S. Rosenbloom, and C. Kommers. Outperforming word2vec on analogy tasks with random projections. *CoRR*, abs/1412.6616, 2014.

[7] L. Derczynski, D. Maynard, G. Rizzo, M. van Erp, G. Gorrell, R. Troncy, J. Petrak, and K. Bontcheva. Analysis of named entity recognition and linking for tweets. *Inf. Process. Manage.*, 51(2):32–49, 2015.

[8] P. Domingos, S. Kok, D. Lowd, H. Poon, M. Richardson, P. Singla, M. Sumner, and J. Wang. Markov logic: A unifying language for structural and statistical pattern recognition. In *SSPR*, 2008.

[9] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *ACM SIGKDD*, 2014.

[10] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In EMNLP, 2011.

[11] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research, 9, 2008.

[12] Y. Goldberg and O. Levy. word2vec explained: deriving mikolov et al.'s negative-sampling wordembedding method. CoRR, abs/1402.3722, 2014.

[13] S. Keretna, C. P. Lim, D. C. Creighton, and K. B. Shaban. Enhancing medical named entity recognition with an extended segment representation technique. Computer Methods and Programs in Biomedicine, 119(2):88–100, 2015.

[14] M. Konkol, T. Brychcin, and M. Konop´ık. Latent semantics in named entity recognition. Expert Syst. Appl., 42(7):3470–3479, 2015.

[15] N. Lao and W. W. Cohen. Relational retrieval using a combination of path-constrained random walks. Machine Learning, 81(1):53–67, 2010.

[16] N. Lao, T. M. Mitchell, and W. W. Cohen. Random walk inference and learning in A large scale knowledge base. In EMNLP, 2011.

[17] S. Lee, H. Lee, P. Abbeel, and A. Y. Ng. Efficient L1 regularized logistic regression. In AAAI, 2006.

[18] V. Leroy, B. B. Cambazoglu, and F. Bonchi. Cold start link prediction. In ACM SIGKDD, 2010.

[19] C. Li, A. Sun, J. Weng, and Q. He. Tweet segmentation and its application to named entity recognition. IEEE Trans. Knowl. Data Eng., 27(2):558–570, 2015.

[20] D. Liben-Nowell and J. M. Kleinberg. The link prediction problem for social networks. In CIKM, 2003.

[21] A. Neelakantan and M. Collins. Learning dictionaries for named entity recognition using minimal supervision. CoRR, abs/1504.06650, 2015.

[22] F. Niu, C. Zhang, C. R´e, and J. W. Shavlik. Elementary: Large-scale knowledge-base construction via machine learning and statistical inference. Int. J. Semantic Web Inf. Syst., 8(3):42–73, 2012.

[23] X. Rong. word2vec parameter learning explained. CoRR, abs/1411.2738, 2014.

[24] T. Shi and Z. Liu. Linking glove with word2vec. CoRR, abs/1411.5595, 2014.

[25] H. H. Song, T. W. Cho, V. Dave, Y. Zhang, and L. Qiu. Scalable proximity estimation and link prediction in online social networks. In ACM SIGCOMM, 2009.

[26] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In WWW, 2007.

[27] B. Suh, G. Convertino, E. H. Chi, and P. Pirolli. The singularity is not near: slowing growth of wikipedia. In Proceedings of the 2009 International Symposium on Wikis, 2009, Orlando, Florida, USA, October 25-27, 2009, 2009.

[28] J. Zhu, Z. Nie, X. Liu, B. Zhang, and J. Wen. Statsnowball: a statistical approach to extracting entity relationships. In WWW, 2009.