Leveraging Wikipedia Knowledge for Entity Recommendations

Nitish Aggarwal*, Peter Mika°, Roi Blanco°, and Paul Buitelaar*

*Insight Centre for Data Analytics
National University of Ireland
Galway, Ireland
firstname.lastname@insight-centre.org
°Yahoo Labs
125 Shaftesbury Ave, WC2H 8HR
London, UK
pmika@yahoo-inc.com, roi@yahoo-inc.com

Abstract. User engagement is a fundamental goal of commercial search engines. In order to increase it, they provide the users an opportunity to explore the entities related to the queries. As most of the queries can be linked to entities in knowledge bases, search engines recommend the entities that are related to the users' search query. In this paper, we present Wikipedia-based Features for Entity Recommendation (WiFER) that combines different features extracted from Wikipedia in order to provide related entity recommendations. We evaluate WiFER on a dataset of 4.5K search queries where each query has around 10 related entities tagged by human experts on 5-level label scale.

1 Introduction

With the advent of large knowledge bases like DBpedia[1], YAGO[2] and Freebase[3], search engines have started recommending entities related to the web search queries. Pound et al. [7] reported that more than 50% web search queries pivot around a single entity and can be linked to an entity in the knowledge bases. Consequently, the task of entity recommendation in the context of web search can be defined as finding the entities related to the entity appearing in a web search query. It is very intuitive to get the related entities by obtaining all the explicitly linked entities to a given entity in knowledge bases. However, most of the popular entities can easily have more than 1,000 directly connected entities, and knowledge bases mainly tend to cover some specific types of relations. For instance, "Tom Cruise" and "Brad Pitt" are not directly connected in the DBpedia graph with any relation, however, they can be considered related to

[1] http://wiki.dbpedia.org/
[2] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/
[3] https://www.freebase.com/

each other as they both are popular Hollywood actors and co-starred in movies. Therefore, to build a system for entity recommendation, there is a need to discover related entities beyond the relations explicitly defined in knowledge bases. Furthermore, these related entities require a ranking method to select the most related ones.

Blanco et al. [4] described the Spark system for related entity recommendation and suggested that such recommendations are successful at extending users' search sessions in Yahoo search. Microsoft also published a similar system [8] that performs personalized entity recommendation by analyzing the click logs. In this paper, we present Wikipedia-based Features for Entity Recommendation (WiFER) that combines different features extracted from Wikipedia. It makes use of Distributional Semantics for Entity Relatedness (DiSER) [1,2] and Explicit Semantic Analysis (ESA) [6] as its features, in combination of others. The features are combined by using learning to rank methods [5]. WiFER is inspired by Spark. However, Spark utilizes proprietary data like query logs and query sessions, which are not available publicly. Therefore, we focus on extracting different features from Wikipedia to build the entity recommendation system.

2 Approach

Wikipedia-based Features for Entity Recommendation (WiFER) combines the different features by using learning to rank method. These features are extracted from Wikipedia by considering two different types of data source: collection of textual content and collection of Wikipedia hyperlinks. The features are derived from the hypothesis that the entities, which occur often in the same context (Wikipedia article), are more likely to be related to each other. We use following features:

1. **Probability (P_1, P_2)** is calculated by taking the ratio of the number of articles that contain the given entity to the total number of articles. P_1 is the probability of an entity E_1. $P_1 = \frac{\sum_{i=0}^{N} o_i}{N}$ where $o_i = 1$, if an article s_i contains the entity E otherwise $o_i = 0$. N is the total number of articles. The value of P of an entity is independent of the other entities, therefore it gives two values P_1 and P_2 for an entity pair consisting of E_1 and E_2.
2. **Joint probability (JPSYM)** This score is obtained by taking the ratio of the number of articles that contain both the given entities to total number of articles. $JPSYM = \frac{\sum_{i=0}^{N} co_i}{N}$ where $co_i = 1$ if an article s_i contains both the entities E_1 and E_2, otherwise $o_i = 0$.
3. **PMI (SISYM)** It computes the point-wise mutual information (PMI). $PMI(E_1, E_2) = \frac{log(P(E_1,E_2))}{P(E_1)*P(E_2))}$ where $P(E_1)$ and $P(E_2)$ are the prior probabilities as described above. $P(E_1, E_2)$ is computed by taking the ratio of number of articles that contain both the entities E_1 and E_2, to the total number of articles.
4. **Cosine similarity (CSSYM)** The cosine similarity is calculated as $Cosine(E_1, E_2) = \frac{P(E_1,E_2)}{P(E_1)*P(E_2))}$

5. **Conditional probability (CPASYM)** It is calculated as the ratio of the total number of articles that contain E_1 and E_2, to the total number of articles that contain E_1. $CPASYM(E_1, E_2) = \frac{\sum_{i=0}^{N} co_i}{\sum_{i=0}^{N} oe_{1i}}$ where $oe_{1i} = 1$ if an article s_i contains the entity E_1, otherwise $oe_{1i} = 0$.
6. **Reverse conditional probability (RCPASYM)** It is reverse of the CPASYM. $RCPASYM(E_1, E_2) = \frac{\sum_{i=0}^{N} co_i}{\sum_{i=0}^{N} oe_{2i}}$ where $oe_{2i} = 1$ if an article s_i contains the entity E_2, otherwise $oe_{1i} = 0$.
7. **Distributional Semantic Model (DSM)** It builds the distributional vector over all the articles [2,6]. DSM computes the values by taking cosine score between the distributional vectors. Therefore, similar to above described features, it relies on the co-occurrence information. However, other features only consider the presence of an entity in the articles and DSM measures the importance of an entity to a given article in addition to its presence.

Since we mentioned that Wikipedia is used twice, WiFER generates 16 different feature values. In order to generate the feature values from text collection, we consider only the surface from of an entity to obtain the occurrence. However, we count the occurrence of an entity in collection of hyperlinks, only if the entity appears as hyperlink in an article. The Probability features generates two values for an entity pair, therefore, each collection provides 8 different feature values and we obtain total 16 values.

3 Evaluation

In order to evaluate our approach, we compare WiFER with the Spark entity recommendation system [4] that uses more than 100 features extracted from different data sources such as query logs and user search sessions. We evaluate the performance on same dataset that was used by Spark. It consists of 4,797 search queries. Every query refers to an entity in DBpedia and contains a list of entity candidates. The entity candidates are tagged by professional editors on 5 label scale: Perfect, Excellent, Good, Fair, and Bad. Finally, it contains 47,623 query-entity pairs. We use Gradient Boosting Decision Tree (GBDT) [5] ranking method. Due to variations in the number of retrieved related entities for a query, we use Normalized Discounted Cumulative Gain (nDCG) for the performance metric. We calculate nDCG@10, nDCG@5, and nDCG@1 as the evaluation metrics. All the nDCG scores are obtained by performing 10-fold cross validation. In addition to performing experiments on the dataset with all the entity types, we also evaluated the systems for the datasets including only person type entities or location type entities. Table 1 shows the retrieval performance of Spark, and compare it with WiFER. It shows that WiFER achieved comparable results on full dataset and person type entities. However, it could not cope well for location type entities. The possible reason behind it could be that most of the locations are too specific which do not have enough information on Wikipedia. Moreover, to investigate if WiFER can complement Spark performance, we combine all the

Features	All			Person			Location		
	ndcg@10	ndcg@5	ndcg@1	ndcg@10	ndcg@5	ndcg@1	ndcg@10	ndcg@5	ndcg@1
Spark	0.9276	0.9038	0.8698	0.9479	0.9337	0.8990	0.8882	0.8507	0.8120
WiFER	0.9173	0.8878	0.8415	0.9432	0.9271	0.8857	0.8795	0.8359	0.7773
Spark+WiFER	0.9325	0.9089	0.8747	0.9505	0.9361	0.9032	0.8987	0.8620	0.8253

Table 1. Retrieval performance on labeled data

features in Spark with WiFER features. WiFER could not outperform Spark, however the combination of both i.e. Spark+WiFER achieved higher scores for all the test cases. Although, WiFER obtained relatively lower scores for location type entities, it is able to compliment the Spark's performance. Further, we performed an extensive evaluation to investigate the importance of different features in entity recommendations (see for more details [3]).

4 Conclusion

In this paper, we presented WiFER that combines different features extracted from Wikipedia, by using a learning to rank method. We showed that WiFER achieved a comparable accuracy to Spark, which uses more than 100 features obtained from proprietary data sources like query logs and user search sessions. Moreover, Spark does not utilize Wikipedia to build its features, thus, we combine WiFER with Spark features, and we showed that WiFER complements the overall performance of Spark.

References

1. N. Aggarwal, K. Asooja, H. Ziad, and P. Buitelaar. Who are the american vegans related to brad pitt?: Exploring related entities. In *Proceedings of the 24th International Conference on World Wide Web Companion*, 2015.
2. N. Aggarwal and P. Buitelaar. Wikipedia-based distributional semantics for entity relatedness. In *2014 AAAI Fall Symposium Series*, 2014.
3. N. Aggarwal, P. Mika, R. Blanco, and P. Buitelaar. Insights into entity recommendation in web search. In *Proceedings of the Intelligent Exploration of Semantic Data, ISWC*, 2015.
4. R. Blanco, B. B. Cambazoglu, P. Mika, and N. Torzec. Entity recommendations in web search. In *International Semantic Web Conference (ISWC)*, 2013.
5. J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.
6. E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 1606–1611, 2007.
7. J. Pound, P. Mika, and H. Zaragoza. Ad-hoc object retrieval in the web of data. In *Proceedings of the 19th international conference on World wide web*, pages 771–780. ACM, 2010.
8. X. Yu, H. Ma, B.-J. P. Hsu, and J. Han. On building entity recommender systems using user click log and freebase knowledge. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 263–272. ACM, 2014.