DBpedia Domains: augmenting DBpedia with domain information

Gregor Titze, Volha Bryl, Cäcilia Zirn, Simone Paolo Ponzetto
Research Group Data and Web Science
University of Mannheim, Germany
firstname@informatik.uni-mannheim.de

Abstract

We present an approach for augmenting DBpedia, a very large ontology lying at the heart of the Linked Open Data (LOD) cloud, with domain information. Our approach uses the thematic labels provided for DBpedia entities by Wikipedia categories, and groups them based on a kernel based k-means clustering algorithm. Experiments on gold-standard data show that our approach provides a first solution to the automatic annotation of DBpedia entities with domain labels, thus providing the largest LOD domain-annotated ontology to date.
Keywords: DBpedia, Wikipedia, domain information

1. Introduction

Recent years have seen a great deal of work on the automatic acquisition of machine-readable knowledge from a wide range of different resources, ranging from the Web (Carlson et al., 2010) all the way to collaboratively-constructed resources used either alone (Bizer et al., 2009b; Ponzetto and Strube, 2011; Nastase and Strube, 2012), or complemented with information from manually-assembled knowledge sources (Navigli and Ponzetto, 2012; Gurevych et al., 2012; Hoffart et al., 2013). The availability of large amounts of high-quality machine readable knowledge, in turn, has led directly to a resurgence of knowledge-rich methods in Artificial Intelligence and Natural Language Processing (Hovy et al., 2013).

All in all, we take this as very good news, since this research trend clearly demonstrates that wide-coverage knowledge bases like DBpedia (Bizer et al., 2009b), YAGO (Hoffart et al., 2013) or BabelNet (Navigli and Ponzetto, 2012) have all the clear potential to yield the next generation of intelligent systems. DBpedia, for instance, has been successfully used for a variety of high-end complex intelligent tasks such as open-domain question answering (Ferrucci et al., 2010), topic labeling (Hulpus et al., 2013), web search result clustering (Schuhmacher and Ponzetto, 2013), and open-domain data mining (Paulheim and Fürnkranz, 2012). However, much still remains to be done to further improve the quality of existing wide-coverage knowledge bases, as well as to extend them with new information. For instance, a kind of information currently missing from any of DBpedia, YAGO or BabelNet is the notion of "domain", namely the fact that concepts and entities in these knowledge bases belong to a set of broad thematic areas, or topics, they are mostly focused on. For instance, information about Barack Obama is mostly about U.S. POLITICS (which, in turn, is a sub-domain of POLITICS), whereas Angela Merkel is mostly focused around GERMAN POLITICS (again, a specialization of the more general topic of POLITICS)[1]. Domain information, in turn, could provide a middle level of abstraction between Wikipedia's concepts and their highly fine-grained categories (e.g., OBAMA being classified as AFRICAN-AMERICAN UNITED STATES PRESIDENTIAL CANDIDATES). In addition, information about domains, such as for instance the one encoded in WordNet Domains (Bentivogli et al., 2004), has been previously shown to benefit important semantic tasks like, for instance, ontology matching (Lin and Sandkuhl, 2008).

In this work we tackle these issues by focusing on the task of discovering the domains of DBpedia entities[2]. We formulate domain typing as the problem of clustering Wikipedia categories associated with each DBpedia entity in a meaningful way, and matching the clusters with the collaboratively-defined domains used to thematically categorize Wikipedia's featured articles[3]. Our results indicate that our method is able to provide a first preliminary solution to the problem of automatically discovering domain types for large amounts of entities in the Linked Open Data cloud.

2. Augmenting DBpedia with domains

We present our method to automatically type DBpedia concepts with domain information. Our approach is based on three main steps:

1. Initially, we collect as topic seeds the information encoded within the categories of the Wikipedia pages associated with DBpedia concepts which, typically, capture very highly-specialized topics. For instance, Barack Obama is associated with many such fine-grained topics (categories) like, among others, the following ones:

   a. Politicians from Chicago, Illinois
   b. African American United States Senators
   c. Democratic Party Presidents of the United States
   d. American Legal Scholars
   e. University of Chicago Law School faculty

[1] Note that more than one label can apply to a specific entity – e.g., people like Arnold Schwarzenegger or Ronald Reagan could be associated with both U.S. POLITICS and MOVIES.

[2] Hereafter, we use *concepts* and *entities* interchangeably to refer to the core resources of the underlying knowledge base, i.e., DBpedia URIs.

[3] Namely, Wikipedia's community-deemed best articles available at `http://en.wikipedia.org/wiki/Wikipedia:Featured_articles`

2. In the next phase, we cluster these categories, in order to automatically create resource-specific domains. In our example above, we would like to group together categories (a-c) to capture the notion of (U.S.) POLITICS. Similarly, categories (d-e) identify entities and concepts within the domain of LAW.

3. In the final phase, we label the clusters by means of a simple, yet effective method which exploits Wikipedia's category tree in order to collect the categories' generalizations – i.e., super-categories – of each cluster member. For instance, categories (a-c) all have POLITICS OF THE UNITED STATES as super-category, which is accordingly set as (one of) the cluster's main label.

In the following, we first briefly introduce DBpedia, the resource used in our methodology, and then move on to explain each phase in details.

2.1. DBpedia

DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web as a full-fledged ontology. The key idea behind DBpedia is to parse *infoboxes*, namely property-summarizing tables found within Wikipedia pages, in order to automatically acquire properties and relations about a large amount of entities. These are further embedded within an ontology based on Semantic Web formalisms such as: i) representing data on the basis of the best practices of *linked data* (Bizer et al., 2009a); ii) encoding semantic relations using the Resource Description Framework (RDF), a generic graph-based data model for describing objects and their relationships. Crucial to our method is the fact that each DBpedia entity is associated with a corresponding Wikipedia page from which the RDF triples describing it were extracted. For instance, the entity `dbp:Barack_Obama`[4] corresponds to the entity described by the Wikipedia page `http://en.wikipedia.org/wiki/Barack_Obama`. Wikipedia categories, which provide a fine-grained thematic categorization of the pages – albeit not taxonomic in nature (Ponzetto and Strube, 2011) – are then included in DBpedia by means of RDF triples expressing `dcterms:subject` relations such as the following one:

```
dbp:Barack_Obama    dcterms:subject
cat:United_States_Senators_from_Illinois
```

2.2. Clustering Wikipedia categories

At the heart of our method lies the identification of the thematic domains of DBpedia entities on the basis of clusters made up of their Wikipedia categories. Our hunch here is to view domains as simply made up of a variety of fine-grained topical categories. Accordingly, we view domain discovery as a clustering task where the objective is to automatically group Wikipedia categories based on their strength of thematic association. That is, we would like to have all categories which are primarily about football in one cluster, those about politics in another one, and so on.

We tackle the task of grouping Wikipedia categories by means of statistical clustering techniques. This has several advantages over rule-based, deterministic approaches like, for instance, grouping based on shared subsumption relations in the Wikipedia category tree. First, it works with as little information as that provided by the category labels – e.g., it can be applied also to other resources which do not have any hierarchical organization of the concepts' categories. Second, machine learning algorithms allow us to include a variety of heterogeneous features to model the degree of similarity between Wikipedia categories.

In this work, we opt for kernel k-means, an algorithm which combines the simplicity of k-means with the power of kernels. The k-means algorithm partitions n observations into k clusters by assigning each data point to the cluster with the nearest mean (which is assumed to be a good prototypical description of the cluster). The best means and clusters are found using an iterative procedure in which (a) each observation is assigned to the cluster whose mean is closest to it (as given by the squared Euclidean distance); (b) each cluster's mean is then re-calculated as the centroid of its observations. The algorithm stops when convergence has been reached – i.e., the cluster assignments no longer change. Kernel k-means works in the same way as the original k-means, except that in the calculation of distance a kernel function is used to calculate Euclidean distance (Dhillon et al., 2004).

In our case, a kernel function is a function $K : C \times C \rightarrow \mathbb{R}$ which for all pairs of Wikipedia categories $c_i, c_j \in C$ calculates their similarity as a metric in a Hilbert feature space F. An example of this is the dot product, i. e. $K(c_i, c_j) = \langle \phi(c_i), \phi(c_j) \rangle$, where $\phi(c_j)$ is a fixed mapping from categories to vector representations in F, namely $\phi : C \rightarrow F$. In this work, we consider the following kernel functions:

1) a **simple co-occurrence kernel** to capture the fact that two categories are more similar if they tend to be assigned to the same pages. This consists of a linear kernel of the form $K_{\text{cooc}}(c_i, c_j) = \phi(c_i)'\phi(c_j)$ which represents the number of co-occurring assignments of categories c_i and c_j to an entity in DBpedia. The feature vector of each category c has the form $\phi(c) = (hasCat_1(c), \ldots, hasCat_{|I|}(c)) \in \{0,1\}^{|I|}$ where $hasCat_i(c)$ is a Boolean function indicating the assignment of category c to DBpedia entity $i \in I$ (namely, the set of all DBpedia entities).

2) a variety of **distributional kernels** (Ó Séaghdha and Copestake, 2008). These compare the categories' co-occurrence probability distributions in order to compute their degree of similarity. To this end, different distance functions on probability measures are used, namely the squared Euclidean L_2 and L_1 distances, Jensen-Shannon divergence and the Hellinger distance, which have all been extensively applied in many NLP tasks.

3) a **string kernel** to capture the similarity between the categories' labels. To compute the kernel function, we

[4] We use `cat:` as abbreviation for `http://dbpedia.org/resource/Category:`, `dcterms:` for `http://purl.org/dc/terms/`, and `dbp:` for `http://dbpedia.org/resource/`.

pre-process Wikipedia category labels as follows. We first part-of-speech tag Wikipedia category labels using the Stanford PoS tagger (Toutanova et al., 2003) and retain only nouns. We next remove stopwords and lemmatize on the basis of a finite-state morphological analyzer (Minnen et al., 2001). We finally associate each category with a vector whose elements represent noun lemmas in the category labels and use these to compute cosine similarity (normalized dot product). This corresponds to the standard bag-of-words kernel (Zhang et al., 2006). Since category names consists of short labels, their terms cannot be associated with meaningful weights. Consequently, we explore in the following two simple variants, namely: a) an unweighted version using binary vectors; b) a manual weighting scheme in which common nouns are assigned double weight with respect to all other terms. Besides the cosine similarity kernel, we additionally test the Tanimoto kernel, originally introduced in (Ralaivola et al., 2005), which is the same as the Jaccard similarity coefficient.

4) a **category tree kernel** which computes similarity as a function of the degree of overlap between the set of super-categories dominating each of the two input categories. To this end, we first compute for each category its inverse category hierarchy graph. This consists of all concepts which dominate a Wikipedia category – i.e., those that can be reached by following the super-category relation along the Wikipedia category tree[5] – up to a maximum depth d. We next associate each category with a binary feature vector encoding the list of super-categories found in its inverse category hierarchy graph, and compute similarity using the cosine metric.

We implement our system using the framework provided by RapidMiner, an open-source machine learning toolkit (Mierswa et al., 2006).

2.3. Labeling the clusters

So far, our system grouped Wikipedia categories into different thematic clusters, i.e., domains, based on a variety of similarity metrics. In the next step, we proceed to identify the most appropriate labels for each of these clusters by looking at Wikipedia's category tree. For each category pair c_i, c_j in a cluster domain D, we first compute the least common subsumer; the cluster label is then set to be the most frequent such concept superordinate across all pairs in D. That is, the domain label is the cluster members' superconcept subsuming as many category pairs as possible.

2.4. Assigning domains to DBpedia entities

In the last step, we assign domains to single entities in DBpedia by simply collecting the labels of the clusters their categories occur within, and keeping the labels with $\geq k$ categories of an instance, where k is a predefined threshold.

[5]This corresponds to the semantic relation `skos:broader` in DBpedia.

3. Evaluation

3.1. Experimental setting

The first step in the evaluation of our clustering approach is to create a gold standard. This is not a trivial task due to the size and the cross-domain nature of DBpedia data. In this work, we opt for using the collection of Wikipedia's featured articles as source of data. Featured articles are a collection of Wikipedia's best articles, as determined by its community of editors. As of April 2014, the set of featured articles contains 4201 pages, which is around 0.1% of English Wikipedia. The articles are divided into 30 thematic categories (e.g. LAW, BIOLOGY, VIDEO GAMING, COMPUTING), and some have one or two subcategories (e.g. COMPANIES and HISTORY BIOGRAPHIES subcategories in BUSINESS, ECONOMICS AND FINANCE and HISTORY, respectively).

To build the gold standard, we selected 60 items from the set of featured articles, two for each of the 30 thematic categories. Articles were selected randomly from the set of all featured articles with a number of categories ≥ 10. We then collected all Wikipedia categories from the sampled articles, providing us with a collection of 947 categories in total, 904 of which are unique. This set of categories was annotated manually with 31 labels: 30 thematic categories plus an extra NON-INFORMATIVE label. The latter was assigned to those categories that, according to the annotators, were too broad and did not convey any topical information (e.g. 1975 ESTABLISHMENTS IN THE UNITED STATES or 1926 BIRTHS). In order to quantify the quality of the annotations and the difficulty of the task, a second annotator was asked to annotate a random sample of 124 categories (making up 13.7% of the whole annotated corpus) and the inter-annotator agreement using the kappa coefficient (Fleiss, 1971) was computed. Our annotators achieved an agreement coefficient κ of 0.79, which indicates a high level of agreement.

Before applying clustering, we filtered away the categories that were considered non-informative with respect to an article topic: namely, subcategories of BIRTHS BY YEAR, DEATHS BY YEAR, ESTABLISHMENTS BY COUNTRY AND YEAR, ALUMNI BY UNIVERSITY OR COLLEGE, and some others – cf. the set of Wikipedia categories covered from the "ByMatcher" method of (Ponzetto and Strube, 2011). These categories were also labeled as NON-INFORMATIVE during the annotation. As a results of this, we removed 218 categories, which left us with a gold standard of 686 items.

3.2. Results and discussion

We report the results in Table 1, using a set of standard clustering measures: Jaccard index, Rand Index (RI), F-measure (F1), Adjusted Rand Index (ARI) and Purity. We compare all previously described kernels, namely: i) a simple co-occurrence kernel (Simple); ii) four distributional kernels based on squared L_2 (L2) distance, L_1 (L1) distance, Jensen-Shannon divergence (JSD) and the Hellinger (Hell) distance; iii) the bag-of-words-based cosine and Tanimoto string kernels; iv) a category tree-based kernel (for different values of the depth search parameter d). In addition, we also experiment with combining the four best-performing kernels of each type: to this end, we opt in

		RI	ARI	Jaccard	F1	Purity
	Singleton	**0.9618**	0.0000	0.0000	0.0862	1.0000
	All-in-one	0.0382	0.0000	0.0382	0.0760	0.0789
	Simple co-occurrence	0.9105	0.1330	0.0965	0.4142	0.4183
distributional kernels	**L2**	0.9304	0.2741	0.1824	0.5152	0.5326
	L1	0.9425	0.2892	0.1898	0.5107	0.5344
	JSD	0.9415	0.3031	0.1999	*0.5342*	*0.5615*
	Hell	*0.9476*	*0.3250*	*0.2138*	0.5248	0.5389
string kernels	**Cosine**	0.9020	0.1107	0.0848	0.4067	0.3857
	Cosine (weighted)	0.9084	0.1350	*0.0980*	*0.4335*	*0.4177*
	Tanimoto	*0.9211*	*0.1385*	0.0978	0.4034	0.3887
category tree kernels	$d = 2$	0.9327	0.0848	0.0637	0.3090	0.3251
	$d = 3$	0.9010	0.1014	0.0798	0.3992	0.3990
	$d = 5$	*0.9439*	*0.2783*	*0.1816*	*0.4867*	*0.5172*
	$d = 8$	0.9376	0.1992	0.1310	0.4085	0.4384
	Linear combination (unweighted)	0.9531	**0.3588**	**0.2370**	**0.5675**	**0.5748**

Table 1: Clustering results: i) singleton and all-in-one-cluster baselines; ii) simple co-occurrence kernel; iii) distributional co-occurrence kernels corresponding to four different distance measures; iv) bag-of-words and Tanimoto string kernels; v) category tree overlap kernels for different depth search parameters; vi) linear kernel combination (unweighted sum). Best results for kernel type are italicized. Best overall results are bolded.

this work for a simple linear combination method that uses the unweighted sum of the kernels as the combined kernel (Gönen and Alpaydin, 2011). Finally, as baselines we use two simple clustering schemes, namely putting all categories into the same cluster (All-in-one), as well as assigning each category to a separate cluster (Singleton).

The results show that all kernels outperform both Singleton and All-in-one baselines. Among the different kernel types, distributional kernels achieve the best performance, thus indicating that basic co-occurrence information already provides us with a strong signal for clustering. Overall, string kernels perform poorly: error analysis revealed that this is due to the fact that they can only produce clusters capturing surface-level string similarity (e.g., ITALY INTERNATIONAL FOOTBALLERS, GERMAN INTERNATIONAL FOOTBALLERS, and so on). The results furthermore indicate that looking at the structure provided by the category tree pays off, in that the category tree kernels achieve performance comparable with the distributional ones (with a search depth limited to a maximum of 5 hops). Finally, the best results are obtained by combining the best kernels of each type, thus indicating that the signals provided by each kernel are complementary in nature, and that a better clustering performance can be achieved by integrating similarity measures from heterogeneous sources.

To have a better understanding of the behavior of the clustering algorithm, we manually looked at the output clusters produced by the Hellinger distance-based distributional kernel and by the combination of the four groups of kernels (see the last line of Table 1). Table 2 presents examples of output clusters, their size, most representative topics (as provided by the annotators in the gold standard using the topics of Wikipedia's featured articles as inventory), as well as examples of categories belonging to the cluster. The examples show that in some clusters highly related topics are grouped together, e.g. GEOGRAPHY AND PLACES and LANGUAGE AND LINGUISTICS in cluster 24, or BIOLOGY and HEALTH AND MEDICINE in cluster 10. Note that our method is able to group together also categories which, albeit topically close in a broad sense, are relatively distant in the Wikipedia category tree, e.g. in cluster 10 categories about different, yet related classes such as mammals, fruits and agriculture are successfully clustered together. An example of low-quality grouping is shown instead in cluster 13, which contains BUSINESS, ECONOMICS AND FINANCE and COMPUTING as primary topics. Within the cluster we find, in fact, categories related to IT companies and computing in general. These examples show that the resulting clusters are meaningful, while not as detailed as we ultimately expect. Combining the kernels allowed us to further improve the clustering by grouping together the categories related to EDUCATION: 3 examples of categories given for cluster 9 actually refer to 3 different Wikipedia pages.

4. Conclusions

In this paper, we presented a first attempt to automatically acquire domains for DBpedia by clustering Wikipedia categories using a kernel-based clustering approach. Although further work is still needed to achieve a performance similar to DBpedia's close-to-human levels of quality, our results indicate the feasibility of the task. Future work will explore supervised techniques to complement clustering, as well as evaluate and develop different methods to label the domains and assign them to entities.

5. Acknowledgments

This work was supported in part by the EU FP7 project LOD2 – Creating Knowledge out of Interlinked Data (Ref. No. 257943) – and the Autonomiefonds of the University of Mannheim.

6. References

Bentivogli, L., Forner, P., Magnini, B., and Pianta, E. (2004). Revising the WordNet Domains hierarchy: semantics, coverage and balancing. In *Proceedings of the COLING-04 Workshop on Multilingual Linguistic Ressources*, pages 101–108.

ID	Size	Topic,%	Examples
10 (h)	33	BIOLOGY 54.5% HEALTH AND MEDICINE, 30.3%	MAMMALS_OF_NORTH_AMERICA TROPICAL_AGRICULTURE FRUITS_ORIGINATING_IN_ASIA
13 (h)	15	BUSINESS, ECONOMICS AND FINANCE, 60.0% COMPUTING, 33.3%	APPLE_INC._ACQUISITIONS CLOUD_COMPUTING_PROVIDERS STEVE_JOBS
24 (h)	34	LANGUAGE AND LINGUISTICS, 50.0% GEOGRAPHY AND PLACES, 35.3% POLITICS AND GOVERNMENT, 14.7%	ARABIC-SPEAKING_COUNTRIES_AND_TERRITORIES MEMBER_STATES_OF_LA_FRANCOPHONIE LANGUAGES_OF_INDONESIA
30 (h)	18	VIDEO GAMING, 100.0%	ADVENTURE_GAMES VIDEO_GAMES_BASED_ON_COMICS
9 (c)	10	EDUCATION, 100.0%	FELLOWS_OF_KING'S_COLLEGE,_CAMBRIDGE FELLOWS_OF_MERTON_COLLEGE,_OXFORD FELLOWS_OF_PEMBROKE_COLLEGE,_OXFORD

Table 2: Clustering results: sample clusters with size, representative topics and examples of categories. (h) stands for *Hellinger* distributional kernel, (c) for the *combination* of all kernels.

Bizer, C., Heath, T., and Berners-Lee, T. (2009a). Linked data – the story so far. *International Journal on Semantic Web and Information Systems*.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009b). DBpedia – A crystallization point for the web of data. *Journal of Web Semantics*, 7(3):154–165.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr., E. H., and Mitchell, T. (2010). Toward an architecture for never-ending language learning. In *Proc. of AAAI-10*, pages 1306–1313.

Dhillon, I. S., Guan, Y., and Kulis, B. (2004). Kernel k-means: spectral clustering and normalized cuts. In *Proc. of KDD '04*, pages 551–556.

Ferrucci, D. A., Brown, E. W., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J. M., Schlaefer, N., and Welty, C. A. (2010). Building Watson: An overview of the DeepQA project. *AI Magazine*, 31(3):59–79.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Gönen, M. and Alpaydin, E. (2011). Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268.

Gurevych, I., Eckle-Kohler, J., Hartmann, S., Matuschek, M., Meyer, C. M., and Wirth, C. (2012). UBY – a large-scale unified lexical-semantic resource based on LMF. In *Proc. of EACL-12*, pages 580–590.

Hoffart, J., Suchanek, F. M., Berberich, K., and Weikum, G. (2013). YAGO2: a spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, pages 28–61.

Hovy, E., Navigli, R., and Ponzetto, S. P. (2013). Collaboratively built semi-structured content and Artificial Intelligence: The story so far. *Artificial Intelligence*, 194:2–27.

Hulpus, I., Hayes, C., Karnstedt, M., and Greene, D. (2013). Unsupervised graph-based topic labelling using DBpedia. In *Proc. of WSDM '13*, pages 465–474.

Lin, F. and Sandkuhl, K. (2008). A survey of exploiting WordNet in ontology matching. In Bramer, M., editor, *IFIP AI*, volume 276 of *IFIP*, pages 341–350. Springer.

Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., and Euler, T. (2006). Yale: Rapid prototyping for complex data mining tasks. In *Proc. of KDD '06*, pages 935–940.

Minnen, G., Carroll, J. A., and Pearce, D. (2001). Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.

Nastase, V. and Strube, M. (2012). Transforming Wikipedia into a large scale multilingual concept network. *Artificial Intelligence*, pages 62–85.

Navigli, R. and Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Ó Séaghdha, D. and Copestake, A. (2008). Semantic classification with distributional kernels. In *Proc. of COLING-08*.

Paulheim, H. and Fürnkranz, J. (2012). Unsupervised feature generation from Linked Open Data. In *Proc. of WIMS'12*.

Ponzetto, S. P. and Strube, M. (2011). Taxonomy induction based on a collaboratively built knowledge repository. *Artificial Intelligence*, 175:1737–1756.

Ralaivola, L., Swamidass, S. J., Saigo, H., and Baldi, P. (2005). Graph kernels for chemical informatics. *Neural Network: Special issue on neural networks and kernel methods for structured domains*, 18(8):1093–1110.

Schuhmacher, M. and Ponzetto, S. P. (2013). Exploiting DBpedia for web search results clustering. In *Proc. of AKBC-13*.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of HLT-NAACL-03*, pages 252–259.

Zhang, L., Zhang, D., Simoff, S. J., and Debenham, J. (2006). Weighted kernel model for text categorization. In *Proc. of AusDM '06*, pages 111–114.