Exploiting DBPedia for Web search results clustering

Michael Schuhmacher
Research Group Data and Web Science
University of Mannheim
Mannheim, Germany
michael@informatik.uni-mannheim.de

Simone Paolo Ponzetto
Research Group Data and Web Science
University of Mannheim
Mannheim, Germany
simone@informatik.uni-mannheim.de

ABSTRACT

We present a knowledge-rich approach to Web search result clustering which exploits the output of an open-domain entity linker, as well as the types and topical concepts encoded within a wide-coverage ontology. Our results indicate that, thanks to an accurate and compact semantification of the search result snippets, we are able to achieve a competitive performance on a benchmarking dataset for this task.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; I.2.4 [**Artificial Intelligence**]: Semantic Networks; I.2.7 [**Artificial Intelligence**]: Natural Language Processing

General Terms

Algorithms, Experimentation

Keywords

Natural Language Processing, Semantic Networks, Search result clustering.

1. INTRODUCTION

Recent years have seen a great deal of work on exploiting semantic models for a wide spectrum of applications, ranging from pre-processing tasks like named entity [5] and word sense disambiguation [29], all the way to high-end applications such as question answering [11] and document search [10]. Complementary to this trend, much research efforts have concentrated on the automatic acquisition of machine-readable knowledge on a large scale by mining large repositories of textual data such as the Web [1, 6, *inter alia*], and exploiting collaboratively-constructed resources either directly [30, 3, 31, 23] or by complementing them with manually-assembled knowledge sources [33, 28, 9, 25, 14]. As a result, recent years have seen a renaissance of knowledge-rich approaches for many different Artificial Intelligence and Natural Language Processing (NLP) tasks [16].

This research trend indicates that semantic information and knowledge-intensive approaches are key components for enabling state-of-the-art performance for many NLP tasks. However, much still remains to be done in order to effectively deploy machine-readable knowledge within high-end applications. Many NLP approaches which draw upon document representations, in fact, rely solely on morpho-syntactic information by means of surface-level meaning representations like vector space models [34]. Although more sophisticated models have been proposed – including conceptual [13] and grounded [4] vector spaces – these still do not exploit the relational knowledge encoded within wide-coverage knowledge bases such as YAGO [33] or DBPedia [3]. This kind of knowledge, in turn, has been shown to benefit knowledge-intensive tasks where semantics plays a crucial role [11].

In this paper, we try to tackle these research issues by looking at the problem of clustering short texts from the Web, such as search result snippets, and see whether this Information Retrieval task can benefit from text semantification, as obtained from the output of a state-of-the-art entity linking system, namely DBPedia Spotlight [22]. Our approach uses DBPedia concepts identified in text as seeds to collect topical concept labels for the snippets. These are then used as features to cluster the snippets based on their topical similarity. Thus, key questions that we aim at addressing with this paper are: (i) whether we can use a state-of-the-art entity disambiguation system to semantify Web data, thus linking them to existing wide-coverage knowledge bases like DBPedia[1]; (ii) whether we can leverage topical (e.g., type-level) information provided by the ontological resource, in order to provide a compact representation of the snippets, and use this to capture their semantic similarity. We evaluate our approach within the experimental framework provided by a SemEval-2013 task aimed at the evaluation of Word Sense Disambiguation and induction algorithms for Web search result clustering [26]: our results indicate that clustering compact, topically semantified representations of snippets is indeed able to yield competitive performance for this task.

[1] We use in this work DBPedia as reference ontology, although our method can be used with other wide-coverage knowledge resource and entity linker, e.g., YAGO [33] and AIDA [15].

AKBC 2013 San Francisco, Cal., 27–28 October, 2013

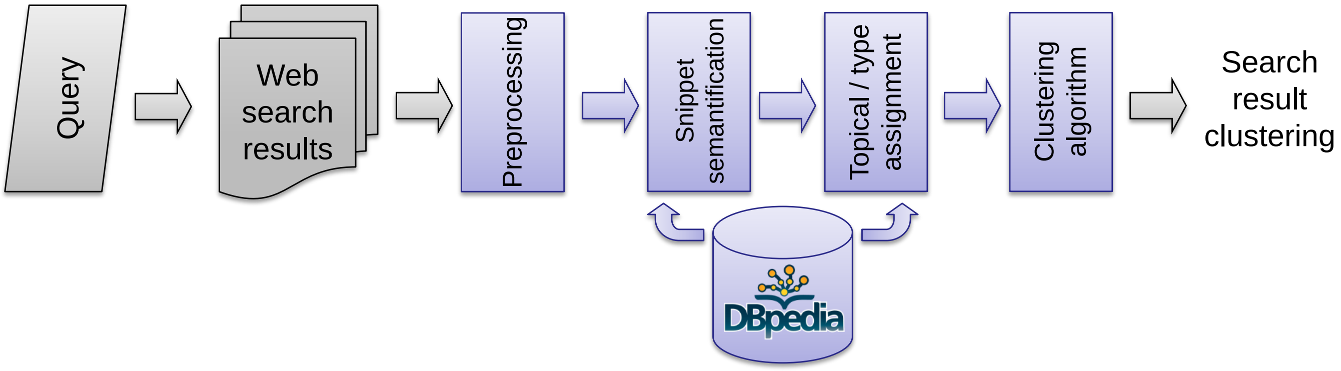

Figure 1: The workflow for our knowledge-rich (i.e., DBPedia-based) approach to Web search result clustering.

2. RELATED WORK

Over the last years many researchers focused on the problem of Web search result clustering – see [7] for a survey. Much work, in particular, has been devoted to identify features which are useful for discriminating the search results' topics, including latent concept models [27], mining query-logs [35], as well as using spectral geometry [20] and graph-clustering algorithms applied to word co-occurrence graphs [24]. The work closest in spirit to ours is that of Scaiella et al. [32], who cluster search results based on a representation of snippets as *graphs of topics*, namely graphs whose nodes correspond to the topics (i.e., Wikipedia pages) identified by applying an entity linker to the texts' snippets. In our work, we also make use of a entity linking system to recognize the most important concepts and entities found within a snippet. However, in contrast to Scaiella et al., we use these concepts to identify the snippets' topics based on their types (as found in DBPedia). We then use these topics as features for a standard clustering algorithm. A limitation of our approach is that it does not exploit structural information for clustering (e.g., the hierarchical relations between the types): however, this allows us to quantify straightforwardly the potential benefit of using only category and type-level information for the task at hand.

3. METHOD

We present a knowledge-rich approach to search result clustering based on the concepts and relations found within a very large ontology, namely DBPedia [3]. Our method takes as input a collection of Web search snippets, and groups them together into topically coherent sets in order to provide the best clustering as output. For instance, given an ambiguous query such as Apache, our dataset contains, among others, the following snippets, which were as returned by the Google search engine [26]:

(1) The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows ...

(2) The Boeing AH-64 Apache is a four-blade, twin-engine attack helicopter with a tailwheel-type landing gear arrangement, and a tandem cockpit for a two-man ...

Each snippet identifies a separate meaning of Apache - namely, the software foundation and the helicopter, in our case. Accordingly, our task is to assign these snippets to different clusters, where each cluster contains snippets conveying the same meaning. We summarize the workflow of our approach in Figure 1. Key to our proposal is (i) a semantified representation of the search result snippets as a bag of the most relevant topical concepts (i.e., types) associated with them, (ii) obtained on the basis of the structure of an underlying ontological resource, i.e., DBPedia. We now turn to describe each component of our system in turn.

Pre-processing. We first pre-process the snippets' text using a standard pipeline of NLP components, including stopword removal and WordNet-based lemmatization, as provided by the NLTK toolkit [2]. Next, we filter out words having a comparably low discriminative power. To this end, we first compute for each word in the snippet a *tf*idf* score using the content of the webpages associated with each snippet. Words in the snippet with a *tf*idf* score below an experimentally determined threshold (as obtained by testing on a development dataset, see Section 4) are excluded from further processing. We perform *tf*idf*-based filtering mainly for two reasons, namely: (a) providing the entity linker with a cleaner, highly discriminant context for disambiguation; (b) removing common words, which could otherwise be annotated with broad, domain-unspecific concepts. Frequency statistics are computed directly from the snippets' documents in order to capture domain-specific usages of words (e.g., Windows being used as a proper name in snippet (1)). As output of this pre-processing step, we end up with snippets containing between 10 and 25 words on average per topic. Given this small size, the corresponding snippets' word vectors are very sparse, and can hardly be used for any similarity computation (which is the basis for snippet clustering). In the next step, we thus aim at acquiring background knowledge capturing the snippets' topics, in order to overcome this sparsity problem.

Snippet semantification. We semantify the snippets by identifying the entities and concepts they are about. To this end, words and phrases are annotated with DBPedia concepts using DBPedia Spotlight [22][2]. Spotlight consists

[2] While there are many entity linking systems which are freely available, in this work we opt for DBPedia Spotlight

of an entity linking system [18] that, given an input text, first identifies mentions collected from Wikipedia anchors, titles and redirects, and found in the DBPedia Lexicalization dataset [21]. Each identified mention is then associated with a set of candidate entities, which define the space of all its possible meanings. Given a mention and its candidate entities, their contexts are represented using a Vector-Space Model (based on a bag-of-words approach), and the candidate whose context has the highest cosine similarity is chosen. Thus, the output of Spotlight consists of a set of disambiguated concepts and entities associated with the words and phrases found in the snippet: for instance, for snippet (1) we are able to establish links to Wikipedia concepts like Apache HTTP Server, HTTP Server, Unix and Microsoft Windows, whereas for snippet (2) we collect Boeing AH-64 Apache, Attack helicopter, Undercarriage, and so on.

Acquiring topical categories of snippets. Spotlight extracts and disambiguates words and phrases by annotating them with unambiguous senses. The resulting DBPedia concepts could, in principle, be used directly as a representation for the snippets. However, questions remain on whether the resulting vectors would be too sparse (as indicated by results on the held-out data observed during prototyping). An alternative would also be to build a bag of words from the text contained within the Wikipedia articles associated with each identified DBPedia concept. However, this surface-level representation would still suffer from the same problems of the simple bag-of-words model, such as not being able, for instance, to capture synonymity – e.g., Wikipedia pages mentioning helicopter and chopper both providing evidence that the snippet belongs to the cluster corresponding the Boeing AH-64 Apache meaning of Apache.

Therefore, we incorporate structured knowledge encoded in DBPedia by retrieving additional concept attributes via the public SPARQL endpoint. We query for all DBPedia and YAGO types denoted by the `rdfs:type` predicate and all Wikipedia categories denoted by the `dcterms:subject` predicate, which have been previously found to provide useful information for topic labeling [17]. As a result, we are able to assign type (from YAGO and DBPedia) and topical (from Wikipedia) labels to all snippets. In our case, for instance, snippet (1) is assigned features such as `dbpedia-owl:Software` and `category:Web_server_software`, whereas snippet (2) is labeled with concepts `dbpedia:Attack_helicopter` and `category:Military_helicopters`, among others. The final snippets' vectors contains only these types and categories, i.e., we leave out the words initially extracted from the snippets. The set of types and categories is thus a document representation by conceptual features, comparable to the Explicit Semantic Analysis approach [13], but created by making use of the explicit semantic relations provided by DBPedia.

Clustering. We finally cluster the snippets using their concept vectors, as obtained in the previous step. To this end, there exists a wide variety of clustering algorithms. In this work, we opt for affinity propagation clustering [12], since it neither requires an a priori fixed number of clusters (like, for instance, k-means), nor it needs a similarity cutoff threshold (in contrast to hierarchical clustering). As standard practice, we manually tune all algorithm-specific parameters such as, for instance, the clustering damping factor, on our held-out data (see Section 4).

System	K			
	5	10	20	40
DWS-Mannheim	37.83	56.31	70.22	83.79
HDP-CLUSTERS-NOLEMMA	50.80	63.21	79.26	92.48
HDP-CLUSTERS-LEMMA	48.13	65.51	78.86	91.68
UKP-WSI-WACKY-LLR	41.19	55.41	68.61	83.90
UKP-WSI-WP-LLR2	41.07	53.76	68.87	85.87
UKP-WSI-WP-PMI	40.45	56.25	68.70	84.92
SATTY-APPROACH1	38.97	48.90	62.72	82.14
DULUTH.SYS7.PK2	38.88	53.79	70.38	86.23
DULUTH.SYS9.PK2	37.15	49.90	68.91	83.65
DULUTH.SYS1.PK2	37.11	53.29	71.24	88.48
RAKESH	46.48	62.36	78.66	90.72

Table 2: S-Recall@K

System	r			
	50	60	70	80
DWS-Mannheim	43.10	32.08	25.72	21.67
HDP-CLUSTERS-LEMMA	48.85	42.93	35.19	27.62
HDP-CLUSTERS-NOLEMMA	48.18	43.88	34.85	29.30
UKP-WSI-WP-PMI	42.83	33.40	26.63	22.92
UKP-WSI-WACKY-LLR	42.47	31.73	25.39	22.71
UKP-WSI-WP-LLR2	42.06	32.04	26.57	22.41
DULUTH.SYS1.PK2	40.08	31.31	26.73	24.51
DULUTH.SYS7.PK2	39.11	30.42	26.54	23.43
DULUTH.SYS9.PK2	35.90	29.72	25.26	21.26
SATTY-APPROACH1	34.94	26.88	23.55	20.40
RAKESH	48.00	39.04	32.72	27.92

Table 3: S-Precision@r

4. EXPERIMENTS

Experimental setting. We evaluate our approach to Web search result clustering on a benchmarking dataset for this task, namely the data from the SemEval-2013 task on 'Evaluating Word Sense Induction & Disambiguation within an End-User Application' [26]. The dataset consists of 100 ambiguous queries (randomly sampled from the AOL search logs) for which there exists a finite set of possible meanings given by a corresponding Wikipedia disambiguation page. Each query comes with 64 search results, as returned by Google's Web search, which are then annotated with any of the meanings provided in the disambiguation page (plus an additional Other class used for snippets for which no sense is appropriate). For system development and parameter tuning, we use Ambient[3], a dataset designed for evaluating subtopic information retrieval, as held-out data.

Results and discussion. We report our results in Table 1, where we evaluate the quality of the clusters output by

since it has been shown to be among the best performing systems on Web text [8].

[3] `http://credo.fub.it/ambient`

System	RI	ARI	JI	F_1	# cl.	ACS
DWS Mannheim (our system)	60.60	9.29	18.70	**69.62**	10.06	9.87
duluth.sys1.pk2	52.18	5.74	31.79	56.83	2.53	26.45
duluth.sys7.pk2	52.04	6.78	31.03	58.78	3.01	25.15
duluth.sys9.pk2	54.63	2.59	22.24	57.02	3.32	19.84
hdp-clusters-lemma	**65.22**	21.31	33.02	68.30	6.63	11.07
hdp-clusters-nolemma	64.86	**21.49**	33.75	68.03	6.54	11.68
satty-approach1	59.55	7.19	15.05	67.09	9.90	6.46
ukp-wsi-wacky-llr	50.02	2.53	**33.94**	58.26	3.64	32.34
ukp-wsi-wp-llr2	51.09	3.77	31.77	58.64	4.17	21.87
ukp-wsi-wp-pmi	50.50	3.64	29.32	60.48	5.86	30.30
rakesh	58.76	8.11	30.52	39.49	9.07	2.94
Singletons	60.09	0.00	0.00	100.00	–	–
All-in-one	39.90	0.00	39.90	54.42	–	–

Table 1: Evaluation results on cluster quality.

our method, as defined in the SemEval task using standard clustering measures from the literature – namely, Rand Index (RI), Adjusted Rand Index (ARI), Jaccard Index (JI) and F_1 measure (F_1). In addition, we report in the table the average number of clusters (# cl.) and average cluster size (ACS) for our system, as well as those which participated to the SemEval task. Finally, we present in Table 2 and 3 our results in the clustering diversity sub-task evaluation – quantified as S-*recall*@K and S-*precision*@r. All performance figures were computed using the SemEval task's official scorer (see [26] for details).

Overall, we generally observe a favorable performance trend, as our system ranks among the best performing ones for this task. In the clustering quality evaluation, in fact, we are able to rank third out of 10 systems in the results of RI and ARI – i.e., right after HDP, the best approach for this task, consisting of a Word Sense Induction system based on Hierarchical Dirichlet Process [19] – and achieve the best F_1 measure overall. Moreover, together with HDP, we are the only system performing above the baseline for RI[4]. Finally, we consistently beat by a large-margin on 3 out of 4 measures Rakesh, the only other knowledge-rich system that participated in the SemEval competition.

When looking at the properties of the clusters themselves (# cl. and ACS) we observe that our approach produces many medium-small sized clusters. We expect this to indicate that, in a Web search result diversification evaluation setting, our system shows a precision-oriented behavior. This analysis is supported by the figures in Table 2 and 3, where we observe that our system generally ranks among the lowest-performing ones in terms of S-Recall@K, whereas it exhibits a middle-level performance on S-Precision@r. The results, thus, seem to indicate that using type-level information from semantified snippets help us focus on more precise meanings of the query terms.

5. CONCLUSIONS

In this paper, we presented a knowledge-based approach to Web search result clustering. Our method exploits the concepts automatically recognized from a state-of-the-art entity linking system and the semantic relations explicitly encoded within a wide-coverage ontology. Our results indicate the viability of using knowledge-rich methods to cluster Web search results beyond the bag-of-words model.

As future work we plan to explore the use of structured representations, i.e., semantic graphs, for this and other related Information Retrieval tasks, as well as exploiting the multilingual dimension encoded within DBPedias from different languages. Finally, we aim at exploring the application of Web search result clustering for ontology population – namely, by extracting domain-specific, updated information from topically-clustered Web text.

[4] As typically the case, baseline methods are notably a difficult competitor for unsupervised and knowledge-rich sense disambiguation and induction systems.

References

[1] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the Web. In *Proc. of IJCAI-07*, pages 2670–2676, 2007.

[2] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python.* O'Reilly Media, 2009.

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

[4] E. Bruni, J. Uijlings, M. Baroni, and N. Sebe. Distributional semantics with eyes: using image analysis to improve computational representations of word meaning. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 1219–1228, 2012.

[5] R. Bunescu and M. Paşca. Using encyclopedic knowledge for named entity disambiguation. In *Proc. of EACL-06*, pages 9–16, 2006.

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

[7] C. Carpineto, S. Osiński, G. Romano, and D. Weiss. A survey of web clustering engines. *ACM Computing Surveys*, 41:17:1–17:38, 2009.

[8] M. Cornolti, P. Ferragina, and M. Ciaramita. A framework for benchmarking entity-annotation systems. In *Proc. of WWW-13*, pages 249–260, 2013.

[9] G. de Melo and G. Weikum. MENTA: inducing multilingual taxonomies from Wikipedia. In *Proc. of CIKM-10*, pages 1099–1108, 2010.

[10] O. Egozi, S. Markovitch, and E. Gabrilovich. Concept-based information retrieval using explicit semantic analysis. *ACM Transactions on Information Systems*, 29(2):8:1–8:34, 2011.

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

[12] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[13] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proc. of IJCAI-07*, pages 1606–1611, 2007.

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

[15] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *Proc. of EMNLP-11*, pages 782–792, 2011.

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

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

[18] H. Ji and R. Grishman. Knowledge base population: Successful approaches and challenges. In *Proc. of ACL-11*, pages 1148–1158, 2011.

[19] J. H. Lau, P. Cook, and T. Baldwin. unimelb: Topic modelling-based word sense induction for web snippet clustering. In *Proc. of SemEval-2013*, pages 217–221, 2013.

[20] Y. Liu, W. Li, Y. Lin, and L. Jing. Spectral geometry for simultaneously clustering and ranking query search results. In *Proc. of SIGIR '08*, pages 539–546, 2008.

[21] P. N. Mendes, M. Jakob, and C. Bizer. DBpedia for NLP: A multilingual cross-domain knowledge base. In *Proc. of LREC-12*, 2012.

[22] P. N. Mendes, M. Jakob, A. Garcia-Silva, and C. Bizer. DBpedia Spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)*, 2011.

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

[24] R. Navigli and A. Di Marco. Clustering and diversifying Web search results with graph-based Word Sense Induction. *Computational Linguistics*, 39(3), 2013.

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

[26] R. Navigli and D. Vannella. Semeval-2013 task 11: Evaluating word sense induction & disambiguation within an end-user application. In *Proc. of SemEval-2013*, pages 193–201, 2013.

[27] S. Osiński and D. Weiss. A concept-driven algorithm for clustering search results. *IEEE Intelligent Systems*, 20(3):48–54, 2005.

[28] S. P. Ponzetto and R. Navigli. Large-scale taxonomy mapping for restructuring and integrating Wikipedia. In *Proc. of IJCAI-09*, pages 2083–2088, 2009.

[29] S. P. Ponzetto and R. Navigli. Knowledge-rich Word Sense Disambiguation rivaling supervised systems. In *Proc. of ACL-10*, pages 1522–1531, 2010.

[30] S. P. Ponzetto and M. Strube. Knowledge derived from Wikipedia for computing semantic relatedness. *Journal of Artificial Intelligence Research*, 30:181–212, 2007.

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

[32] U. Scaiella, P. Ferragina, A. Marino, and M. Ciaramita. Topical clustering of search results. In *Proc. of WSDM '12*, pages 223–232, 2012.

[33] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A large ontology from Wikipedia and WordNet. *Journal of Web Semantics*, 6(3):203–217, 2008.

[34] P. D. Turney and P. Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188, 2010.

[35] X. Wang and C. Zhai. Learn from web search logs to organize search results. In *Proc. of SIGIR '07*, pages 87–94, 2007.