GrawlTCQ: Terminology and Corpora Building by Ranking Simultaneously Terms, Queries and Documents using Graph Random Walks

Clément de Groc
Syllabs
Univ. Paris Sud
LIMSI-CNRS
cdegroc@limsi.fr

Xavier Tannier
Univ. Paris Sud
LIMSI-CNRS
xtannier@limsi.fr

Javier Couto
Syllabs
MoDyCo (UMR 7114, CNRS-UPX)
jcouto@syllabs.com

Abstract

In this paper, we present GrawlTCQ, a new bootstrapping algorithm for building specialized terminology, corpora and queries, based on a graph model. We model links between documents, terms and queries, and use a random walk with restart algorithm to compute relevance propagation. We have evaluated GrawlTCQ on an AFP English corpus of 57,441 news over 10 categories. For corpora building, GrawlTCQ outperforms the BootCaT tool, which is vastly used in the domain. For 1,000 documents retrieved, we improve mean precision by 25%. GrawlTCQ has also shown to be faster and more robust than BootCaT over iterations.

1 Introduction

Specialized terminology and corpora are key resources in applications such as machine translation or lexicon-based classification, but they are expensive to develop because of the manual validation required. Bootstrapping is a powerful technique for minimizing the cost of building these resources.

In this paper, we present GrawlTCQ[1], a bootstrapping algorithm for building specialized terminology, corpora and queries: from a small set of user-provided terms, GrawlTCQ builds the resources via automated queries to a search engine. The algorithm relies on a graph that encodes the three kinds of entities involved in the procedure (*terms*, *documents* and *queries*) and relations between them. We model the relevance propagation in our graph by using a random walk with restart algorithm.

We use BootCaT (Baroni and Bernardini, 2004) as our baseline because it is a similar algorithm that has been vastly used and validated experimentally in the domain. We have evaluated GrawlTCQ and BootCaT on an AFP (Agence France Presse) English corpus of 57,441 news over 10 categories. Results show that, for corpora building, GrawlTCQ significantly outperforms the BootCaT algorithm. As this is an on-going work, further work is needed to evaluate terminology and query results.

The article is structured as follows: in Section 2, we review the related work in terminology and corpora construction using bootstrapping techniques, as well as random walk applications. In Section 3, we describe GrawlTCQ. In Section 4, we evaluate GrawlTCQ and compare its results with those provided by BootCaT. We conclude in Section 5.

2 Related Work

Several works using bootstrapping techniques have been carried out in terminology and corpora creation. For example, (Ghani et al., 2005) has built minority language corpora from the web. The Web-as-Corpus WaCky initiative (Baroni et al., 2009; Ferraresi et al., 2008; Sharoff, 2006) has built very large web-derived corpus in various languages. They used previously mentioned BootCaT tool to do this. As the quality of the results is strongly dependent on the quality of seed terms and the underlying search engine, manual filtering is usually mandatory to enhance performance. GrawlTCQ uses a graph to automatically filter out erroneous terms and documents

[1] GrawlTCQ stands for Graph RAndom WaLk for Terminology, Corpora and Queries.

Proceedings of the TextGraphs-6 Workshop, pages 37–41,
Portland, Oregon, USA, 19-24 June 2011.

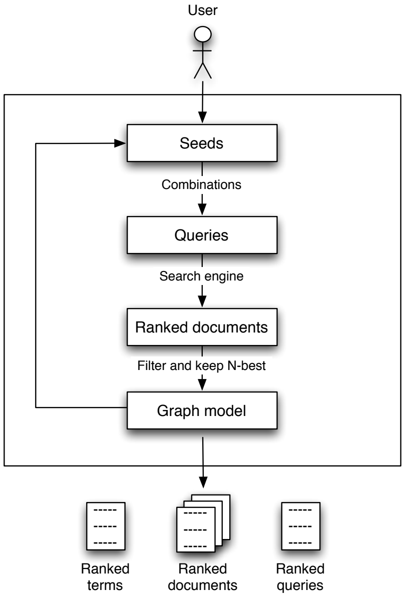

Figure 1: Components of the GrawlTCQ algorithm.

and improve the system's overall performance. The manual filtering cost is therefore drastically reduced.

Graph modeling and random walks have been applied with success to many different domains of NLP, such as keyword and sentence extraction (Mihalcea and Tarau, 2004), computer-science articles ranking (Nie et al., 2005), web pages ranking (Haveliwala, 2002; Page et al., 1999; Richardson and Domingos, 2002), WordNet-based word sense disambiguation (Agirre and Soroa, 2009) and lexical semantic relatedness (Hughes and Ramage, 2007), or set expansion (Wang and Cohen, 2007). In this paper, we confirm the relevance of this approach to terminology and corpora bootstrapping.

3 Ranking simultaneously Terms, Queries and Documents

3.1 The GrawlTCQ bootstrapping algorithm

Figure 1 shows the components of the GrawlTCQ algorithm. Starting from user provided seed terms[2], GrawlTCQ iteratively creates queries, finds documents and extracts new terms. We model this bootstrapping procedure with a graph that keeps all links between documents, terms and queries. Our hypothesis is that the information added will increase the procedure's robustness and overall performances. The graph model (see figure 2) is built online. As common terms will occur in many documents and thus have high centrality, they will end with high scores. In order to avoid this effect, document-term edges are weighted with a TermHood measure (Kageura and Umino, 1996) such as tfidf or log odds ratio.

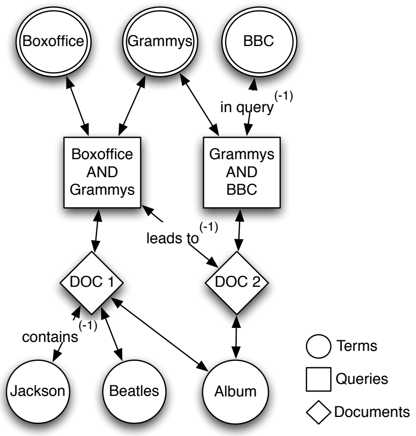

Figure 2: Sample subgraph using ”boxoffice”, ”Grammys” and ”BBC” as seed terms.

By using a random walk with restart algorithm, also known as personalized PageRank (Haveliwala, 2002), terms, queries and documents are weighted globally and simultaneously. At the end of each iteration of GrawlTCQ, a random walk is computed and the resulting stationary distribution is used to rank documents and terms[3]. If more documents are needed, then the algorithm executes one more step.

Several parameters can be specified by the user, such as the number of seed terms, the number of terms composing a query, as well as the number of documents retrieved for each query. In addition, the algorithm may use the Internet (with search engines as Google, Yahoo! or Bing), an Intranet, or both, as data sources. When using the web as source, specific algorithms must be used to remove HTML boilerplate (Finn et al., 2001) and filter un-useful documents (duplicates (Broder, 2000), webspam and error pages (Fletcher, 2004)).

[2] These terms may be easily computed from a list of seed urls or documents, using terminology extraction techniques.

[3] As an additional result, we also obtain a ranked list of queries.

3.2 Graph Walk

Considering a directed graph $G = (V, E)$, the score of a vertex V_i is defined as

$$PR(V_i) = (1 - \alpha)\lambda_0 + \alpha \times \sum_{j \in In(V_i)} \frac{PR(V_j)}{|Out(V_j)|}$$

where $In(V_i)$ (resp. $Out(V_i)$) are V_i predecessors (resp. successors). In the original PageRank algorithm, a damping factor α of 0.85 has been used and the personalization vector (or teleportation vector) λ_0 is distributed uniformly over V. On the contrary, (Richardson and Domingos, 2002) and (Haveliwala, 2002) have proposed to personalize the PageRank according to a user query or a chosen topic. Following previous work (Page et al., 1999; Mihalcea and Tarau, 2004), we have fixed the damping factor to 0.85[4] and the convergence threshold to 10^{-8}.

As we have different types of edges carrying different relations, we slightly modify the PageRank formula, as in (Wang and Cohen, 2007): when walking away from a node, the random surfer first picks randomly a relation type and then chooses uniformly between all edges of the chosen relation type. Biasing the algorithm to insist more on seed terms is a legitimate lead as these nodes represent the strong base of our model. We thus use a custom λ_0 distribution that spreads weights uniformly over the seed terms instead of the whole set of vertices.

4 Evaluation

Evaluating the proposed method on the web can hardly be done without laborious manual annotation. Moreover, web-based evaluations are not reproducible as search engines index and ranking functions change over time. This is especially a problem when evaluating the impact of different parameters of our algorithm. In this article, we have chosen to carry out an objective and reproducible evaluation based on a stable and annotated document collection.

The AFP has provided us an English corpus composed of 57,441 news documents written between January 1st and March 31, 2010. We have considered the 17 top-level categories from the IPTC standard (http://www.iptc.org). Documents are categorized in one or more of those categories and are annotated with various metadata, such as keywords. As some categories contained too few documents, we have only kept the 10 largest ones (see table 1). The corpus was then indexed using Apache Lucene (http://lucene.apache.org) in order to create a basic search engine[5]. This setup has several advantages: first, the document collection is stable and quantifiable. Documents are clean text written in a journalistic style. As they are already annotated, several automatic evaluations can be run with different parameters. Finally, querying the search engine and retrieving documents can be done efficiently. However, note that, as the document collection is limited, queries might return few or no results (which is rarely the case on the web).

Id	Category	#docs
01	Arts, culture and entertainment	3074
02	Crime, law and justice	5675
03	Disaster and accident	4602
04	Economy, business and finance	13321
08	Human interest	1300
11	Politics	17848
12	Religion and belief	1491
14	Social issue	1764
15	Sport	15089
16	Unrest, conflicts and war	8589

Table 1: AFP corpus categories distribution.

We have used the BootCaT algorithm as our baseline. To the best of our knowledge this is the first attempt to rigorously evaluate BootCaT performances. We have compared both algorithms in exactly the same conditions, on a task-based experiment: to retrieve 50, 100, 300, 500 and 1000 documents for each category, independently of the number of iterations done.

To be as close as possible to the original BootCaT algorithm, we have weighted document-term edges by log odds ratio. This measure allows us to distinguish common terms by using a reference background corpus. In all our experiments, we have used the ukWac corpus (Ferraresi et al., 2008), a very large web-derived corpus, for this purpose.

In order to select initial seed terms we have used documents' metadata. We have computed the fre-

[4] During our experiments, we haven't observed any significant change when modifying this parameter.

[5] All normalization features except lower-casing were disabled to allow ease of reproducibility.

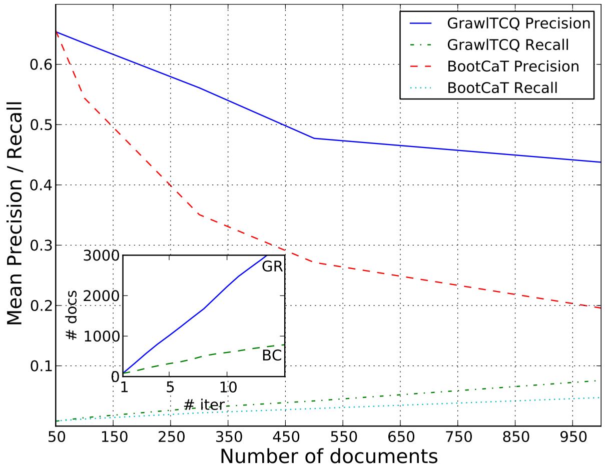

Figure 3: Mean precision and recall at 50, 100, 300, 500 and 1000 documents (inset: Mean number of documents / number of iterations)

quency of occurrences of a keyword in a category and have then divided this score by the sum of occurrences in all other categories. This strategy leads to relevant seed terms that are not necessarily exclusive to a category. For instance, selected seeds for the 4th category are: *economics*, *summary*, *rate*, *opec*, *distress*, *recession*, *zain*, *jal*, *gold*, and *spyker*.

We have fixed a number of parameters for our experiments: at each iteration, the top-10 seeds are selected (either from the initial set or from newly extracted terms). Queries are composed of 2 seeds, all 45 possible combinations[6] are used and a total of 10 documents are retrieved for each query.

CatId	P@50		P@100		P@300		P@500		P@1000	
	GR	BC	GR	BC	GR	BC	GR	BC	GR	BC
01	**0.58**	0.50	**0.57**	0.30	**0.43**	0.12	**0.35**	0.08	**0.23**	0.05
02	0.44	**0.60**	**0.45**	0.33	**0.46**	0.17	**0.44**	0.10	**0.34**	0.07
03	0.82	0.82	**0.99**	0.81	**0.89**	0.41	**0.66**	0.26	**0.54**	0.14
04	**0.86**	0.80	0.82	**0.85**	**0.84**	0.55	**0.78**	0.34	**0.79**	0.19
08	**0.79**	0.79	0.44	**0.48**	0.23	**0.42**	0.17	**0.40**	0.20	**0.39**
11	0.76	**0.78**	0.79	**0.81**	**0.87**	0.71	0.57	**0.64**	**0.57**	0.56
12	0.46	**0.54**	**0.35**	0.27	**0.20**	0.10	**0.17**	0.06	**0.15**	0.03
14	0.08	**0.24**	**0.13**	0.10	**0.06**	0.04	**0.04**	0.02	**0.04**	0.02
15	1.0	1.0	1.0	1.0	**0.92**	0.78	**0.87**	0.67	**0.81**	0.39
16	**0.82**	0.56	**0.81**	0.49	**0.71**	0.21	**0.72**	0.15	**0.70**	0.13

Table 2: Precision at various cutoffs by category

All scores are averaged over the 10 categories. As can be seen in figure 3, GrawlTCQ shows much more robustness and outperforms BootCaT by 25% precision at 1000 documents. Detailed results for each category are shown in table 2 and confirm the relevance of our approach. Interestingly, BootCaT and GrawlTCQ have very low precisions for the 14th category (*Social issue*). Documents found in this category are often ambiguous and both algorithms fail to extract the domain terminology. We have also plotted the number of documents in function of the number of iterations as shown in figure 3 (inset). The curve clearly shows that GrawlTCQ yields more documents at a faster rate. This is due to the seed selection process: GrawlTCQ's queries lead to many documents while BootCaT queries often lead to few or no documents. Moreover, as we can see in figure 3, while fetching more documents faster, the mean precision of GrawlTCQ is still higher than the BootCaT one which shows that selected seeds are, at the same time, more prolific and more relevant.

5 Conclusion

In this paper, we have tackled the problem of terminology and corpora bootstrapping. We have proposed GrawlTCQ, an algorithm that relies on a graph model including terms, queries, and documents to track each entity origin. We have used a random walk algorithm over our graph in order to globally and simultaneously compute a ranking for each entity type. We have evaluated GrawlTCQ on a large news dataset and have shown interesting gain over the BootCaT baseline. We have especially obtained better results without any human intervention, reducing radically the cost of manual filtering. We are considering several leads for future work. First, we must evaluate GrawlTCQ for query and term ranking. Then, while preliminary experiments have shown very promising results on the web, we would like to setup a large scale rigorous evaluation. Finally, we will conduct further experiments on edges weighting and seed terms selection strategies.

Acknowledgments

We would like to thank the AFP for providing us the annotated news corpus. This work was partially funded by the ANR research project ANR-08-CORD-013.

[6] When running the same experiment with randomly selected tuples several times, we have found similar results when averaging all runs output.

References

Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics on - EACL, 2009*.

Marco Baroni and Silvia Bernardini. 2004. BootCaT: Bootstrapping Corpora and Terms from the Web. In *Proceedings of the LREC 2004 conference*.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky Wide Web : A Collection of Very Large Linguistically Processed Web-Crawled Corpora. In *Proceedings of the LREC 2009 conference*, volume 43, pages 209–226.

Andrei Z Broder. 2000. Identifying and Filtering Near-Duplicate Documents. In *Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching*, pages 1–10, London, UK. Springer-Verlag.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4)*, pages 47–54.

Aidan Finn, Nicholas Kushmerick, and Barry Smyth. 2001. Fact or fiction: Content classification for digital libraries. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*.

William H Fletcher. 2004. Making the Web More Useful as a Source for Linguistic Corpora. *Corpus Linguistics in North America*, (January 2003):191–205.

Rayid Ghani, Rosie Jones, and Dunja Mladenic. 2005. Building Minority Language Corpora by Learning to Generate Web Search Queries. *Knowl. Inf. Syst.*, 7(1):56–83.

Taher H. Haveliwala. 2002. Topic-sensitive PageRank. *Proceedings of the eleventh international conference on World Wide Web - WWW '02*, page 517.

Thad Hughes and Daniel Ramage. 2007. Lexical Semantic Relatedness with Random Graph Walks. In *Proceedings of EMNLP, 2007*, pages 581–589.

Kyo Kageura and Bin Umino. 1996. Methods of automatic term recognition: A review. *Terminology*, 3(2):259–289.

Rada Mihalcea and Paul Tarau. 2004. TextRank bringing order into text. In *Proceedings of EMNLP*, pages 404–411. Barcelona: ACL.

Zaiqing Nie, Yuanzhi Zhang, J.R. Wen, and W.Y. Ma. 2005. Object-level ranking: Bringing order to web objects. In *Proceedings of the 14th international conference on World Wide Web*, pages 567–574. ACM.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford InfoLab.

M. Richardson and P. Domingos. 2002. The intelligent surfer: Probabilistic combination of link and content information in pagerank. *Advances in Neural Information Processing Systems*, 2:1441–1448.

Serge Sharoff. 2006. Creating general-purpose corpora using automated search engine queries. *M. Baroni, S. Bernardini (eds.) WaCky! Working papers on the Web as Corpus, Bologna, 2006*, pages 63–98.

Richard C. Wang and William W. Cohen. 2007. Language-independent set expansion of named entities using the web. *Proceedings of IEEE International Conference on Data Mining (ICDM 2007), Omaha, NE, USA. 2007.*