Language Evolution On The Go

Gideon Zenz, Nina Tahmasebi and Thomas Risse

L3S Research Center, Hannover, Germany,
{zenz, tahmasebi, risse}@L3S.de

Abstract. Knowing about the evolution of a term can significantly decrease time needed for searching for information. It can also aid in quickly getting a broader overview, which is essential when one is on the move. In this paper we present a solution for providing language evolution knowledge "on the go". We present a mobile interface for easy access and visualization as well as an overview of how this evolution was found.

Keywords: language evolution, mobile applications, ambient media

1 Introduction

In the age of ambient media, the user demands constant support for her needs. Mobile and smart devices provide excellent facilities for giving immersive, location based support for activities. In this paper we present a solution for providing language evolution knowledge "on the go".

Languages are evolving over time triggered by various factors including new insights, political and cultural trends, new legal requirements, and high-impact events [2]. Imagine traveling through St. Petersburg. This city exhibits a particularly interesting language development, as it was founded in 1703 as "Sankt Piter Burh" and soon after renamed to "Saint Petersburg". From 1914-1924 it was named "Petrograd" and afterwards "Leningrad". In 1991 it changed back to "Saint Petersburg" also simply referred to as "Petersburg". The typical user will not be aware of this complex development and therefore might be puzzled by observing different names when sightseeing. Using our terminology evolution application, such connections can be more easily and conveniently determined than using standard search on e.g. Google or Wikipedia.

To our knowledge only one previous work has been published in the area of terminology evolution[1]. Using language from the past, the aim here is to find good query reformulations for search engines of concurrent language. Our approach advances on this by using word senses to find similar terms rather than pure co-occurrence information. Furthermore our approach does not restrict the user to specifying a timeframe for the evolution. Due to the limited previous work no investigations on the interaction with user for this special application have been conducted.

The contribution of this paper is the development of an initial mobile interface for easy access and visualization of the language evolution we detected in a

large real-world corpus - The Times Archive[1] - for the usage on mobile devices like iPads or WebPads.

In the following we will first give some background information on the detection of language evolution. Afterwards in Section 3 we present our user interface of our language evolution application. Finally we conclude and give an outlook on future work.

2 Terminology Evolution

The challenges in language or terminology evolution are broad and cover the detection of added, removed and changed senses for a word. It also includes different terms for the same concept like in the St. Petersburg example. By comparing found word senses over time, important information can be revealed.

2.1 Finding word senses

The first step of detecting language evolution is to automatically detect word senses given a text collection. For this we use a word sense discrimination algorithm know as *curvature clustering* [3]. Following we describe the steps involved.

Natural Language Processing First the text is cleaned from strange tokens. The text is lemmatized and identified nouns and noun phrases are added to a term list which is considered to be the *dictionary* corresponding to the collection.

Co-Occurrence Graph Creation We create a *co-occurrence graph* using the dictionary. The collection is searched for lists of nouns and noun phrases. Terms from the dictionary, that are found separated by an "and", an "or" or a comma, are considered co-occurring. E.g., in "*... cities such as Paris, New York and Berlin ...*" the terms "*Paris*", "*New York*" and "*Berlin*" all co-occur in the graph. Once the entire collection is processed, all co-occurrences are filtered and low frequency co-occurrences are removed to reduce the level of noise.

Graph Clustering To cluster the graph, we use the curvature clustering algorithm which calculates the clustering coefficient [5], curvature value, of each node. After computing the curvature values, the algorithm removes nodes with a curvature value below a certain threshold. The low curvature nodes represent ambiguous nodes that are likely to connect parts of the graph that would otherwise not be connected. Once these nodes are removed, the remaining graph falls apart into connected components, from now on referred to as clusters. Clusters are considered to be candidate word senses. Finally each cluster is enriched with the nearest neighbors of its members to capture ambiguity.

[1] http://archive.timesonline.co.uk/tol/archive/

2.2 Finding word sense evolution

The second step for finding terminology evolution is to track the word senses over time. The tracking is done for each term separately. We compare the clusters where the term participates to see if there has been any evolution. Current tracking technology use Jaccard similarity to compare clusters. The similarity scores for two clusters lie between 0 and 1 where 1 indicates that two clusters are exactly the same and 0 indicates no terms in common. We consider two clusters which have a similarity higher than α to represent the same word sense. When two clusters have a similarity below β we consider the clusters to have no relation. Clusters with similarity above β but below α are candidates for evolution.

3 User Interface and implementation

In order to make the results of the language evolution process end-user accessible, we devised a mobile web service which allows for exploring the evolution of a given term. As running example we will use the term *Petersburg* present in clusters extracted from The Times Archive (1785-1985) [4]. After the user specifies the term of interest, we show all clusters containing this term over time. As representative, we chose the term with the highest clustering coefficient (on top, Figure 1). Furthermore, we give the term frequency distribution of the term over time (bottom, Figure 1).

By assessing the term frequency distribution, and possibly combined with a changing cluster representative as seen on the right side, the user can infer if a significant change of the word usage happened at a given point in time.

To get a deeper understanding of the context of a given year, the user can simply touch a cluster representative to see all cluster members along with their connection.

The terminology evolution application enables the user to get a quick look at what happens to a term over time. First of all the raw (or normalized) term frequencies over time can give an indication of an event, or evolution, for a term. When the term "Petersburg" looses in frequency from 1914 to 1915 it is worth to investigate further into that point in time. In this example it corresponds to when "St. Petersburg" changes name to "Petrograd". In addition to the term frequencies, the clusters help with understanding the term context.

The terminology evolution application saves the user time in finding and understanding the context of a term from different periods in time and shows semantic relations, which are otherwise much more complicated to obtain using standard Google, or Wikipedia searches.

4 Conclusions

In this paper, we presented a solution for providing language evolution "on the go". As a basis we used The Times Archive, a large real-world corpus, allowing us

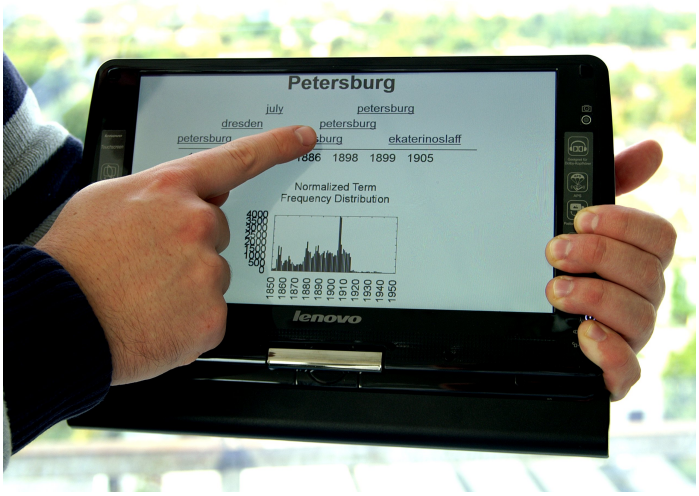

Fig. 1. User-interface showing clusters for term *Petersburg*

to identify significant evolutions in language. We devised an application tailored for mobile devices, which allows for easy access to language evolution "on the go". Initial results suggest a significant improvement over standard knowledge accessing mechanisms. Future work is a formal user study to further improve the application.

5 Acknowledgments

We would like to thank Times Newspapers Limited for providing the archive of The Times for our research.

References

1. K. Berberich, S. Bedathur, M. Sozio, and G. Wiekum. Bridging the terminology gap in web archive search. In *WebDB*, 2009.
2. M. C. Cooper. A mathematical model of historical semantics and the grouping of word meanings into concepts. *Computational Linguistics*, 32(2):227–248, 2005.
3. B. Dorow, J. pierre Eckmann, and D. Sergi. Using curvature and markov clustering in graphs for lexical acquisition and word sense discrimination. In *In Workshop MEANING-2005*, 2004.
4. N. Tahmasebi, K. Niklas, T. Theuerkauf, and T. Risse. Using word sense discrimination on historic document collections. In *JCDL '10: Proceedings of the 10th ACM/IEEE-CS joint conference on Digital libraries*, Gold Coast, Australia, 2010. ACM.
5. D. Watts and S. Strogatz. Collective dynamics of "small-world" networks. *Nature*, 393:440–442, 1998.