A Mobile Touchable Application for Online Topic Graph Extraction and Exploration of Web Content

Günter Neumann and Sven Schmeier
Language Technology Lab, DFKI GmbH
Stuhlsatzenhausweg 3, D-66123 Saarbrücken
{neumann|schmeier}@dfki.de

Abstract

We present a mobile touchable application for online topic graph extraction and exploration of web content. The system has been implemented for operation on an iPad. The topic graph is constructed from N web snippets which are determined by a standard search engine. We consider the extraction of a topic graph as a specific empirical collocation extraction task where collocations are extracted between chunks. Our measure of association strength is based on the pointwise mutual information between chunk pairs which explicitly takes their distance into account. An initial user evaluation shows that this system is especially helpful for finding new interesting information on topics about which the user has only a vague idea or even no idea at all.

1 Introduction

Today's Web search is still dominated by a document perspective: a user enters one or more keywords that represent the information of interest and receives a ranked list of documents. This technology has been shown to be very successful when used on an ordinary computer, because it very often delivers concrete documents or web pages that contain the information the user is interested in. The following aspects are important in this context: 1) Users basically have to know what they are looking for. 2) The documents serve as answers to user queries. 3) Each document in the ranked list is considered independently.

If the user only has a vague idea of the information in question or just wants to explore the information space, the current search engine paradigm does not provide enough assistance for these kind of searches. The user has to read through the documents and then eventually reformulate the query in order to find new information. This can be a tedious task especially on mobile devices. Seen in this context, current search engines seem to be best suited for "one-shot search" and do not support content-oriented interaction.

In order to overcome this restricted document perspective, and to provide a mobile device searches to "find out about something", we want to help users with the web content exploration process in two ways:

1. We consider a user query as a specification of a topic that the user wants to know and learn more about. Hence, the search result is basically a graphical structure of the topic and associated topics that are found.

2. The user can interactively explore this topic graph using a simple and intuitive touchable user interface in order to either learn more about the content of a topic or to interactively expand a topic with newly computed related topics.

In the first step, the topic graph is computed on the fly from the a set of web snippets that has been collected by a standard search engine using the initial user query. Rather than considering each snippet in isolation, all snippets are collected into one document from which the topic graph is computed. We consider each topic as an entity, and the edges

between topics are considered as a kind of (hidden) relationship between the connected topics. The content of a topic are the set of snippets it has been extracted from, and the documents retrievable via the snippets' web links.

A topic graph is then displayed on a mobile device (in our case an iPad) as a touch-sensitive graph. By just touching on a node, the user can either inspect the content of a topic (i.e, the snippets or web pages) or activate the expansion of the graph through an on the fly computation of new related topics for the selected node.

In a second step, we provide additional background knowledge on the topic which consists of explicit relationships that are generated from an online Encyclopedia (in our case Wikipedia). The relevant background relation graph is also represented as a touchable graph in the same way as a topic graph. The major difference is that the edges are actually labeled with the specific relation that exists between the nodes.

In this way the user can explore in an uniform way both new information nuggets and validated background information nuggets interactively. Fig. 1 summarizes the main components and the information flow.

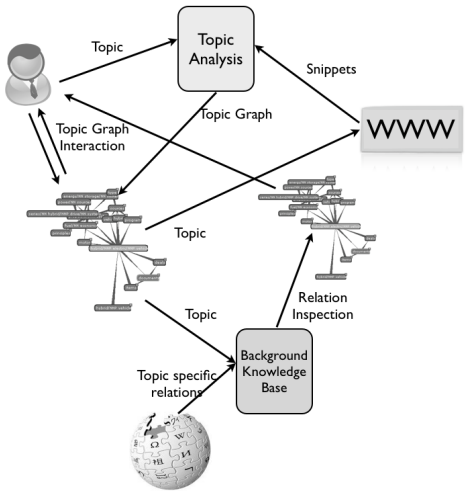

Figure 1: Blueprint of the proposed system.

2 Touchable User Interface: Examples

The following screenshots show some results for the search query "Justin Bieber" running on the current iPad demo–app. At the bottom of the iPad screen, the user can select whether to perform text exploration from the Web (via button labeled "i–GNSSMM") or via Wikipedia (touching button "i–MILREX"). The Figures 2, 3, 4, 5 show results for the "i–GNSSMM" mode, and Fig. 6 for the "i-MILREX" mode. General settings of the iPad demo-app can easily be changed. Current settings allow e.g., language selection (so far, English and German are supported) or selection of the maximum number of snippets to be retrieved for each query. The other parameters mainly affect the display structure of the topic graph.

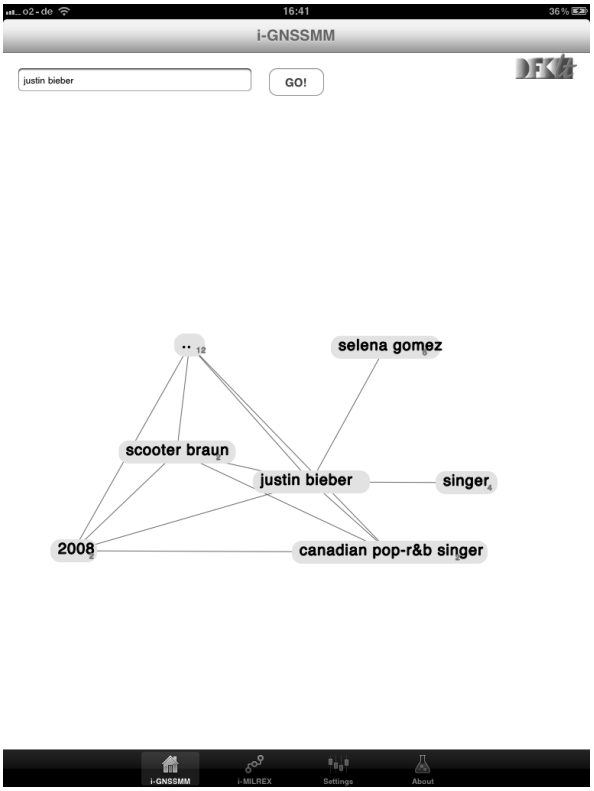

Figure 2: The topic graph computed from the snippets for the query "Justin Bieber". The user can double touch on a node to display the associated snippets and web pages. Since a topic graph can be very large, not all nodes are displayed. Nodes, which can be expanded are marked by the number of hidden immediate nodes. A single touch on such a node expands it, as shown in Fig. 3. A single touch on a node that cannot be expanded adds its label to the initial user query and triggers a new search with that expanded query.

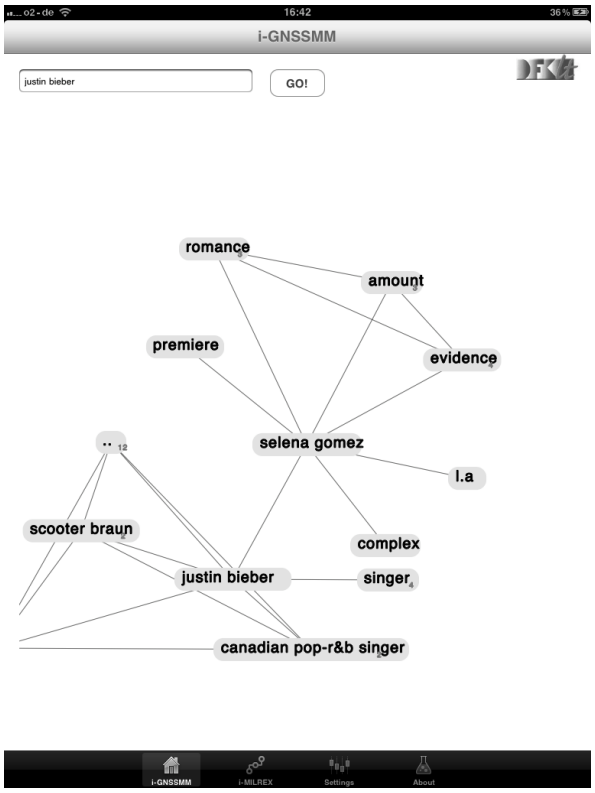

Figure 3: The topic graph from Fig. 2 has been expanded by a single touch on the node labeled "selena gomez". Double touching on that node triggers the display of associated web snippets (Fig. 4) and the web pages (Fig. 5).

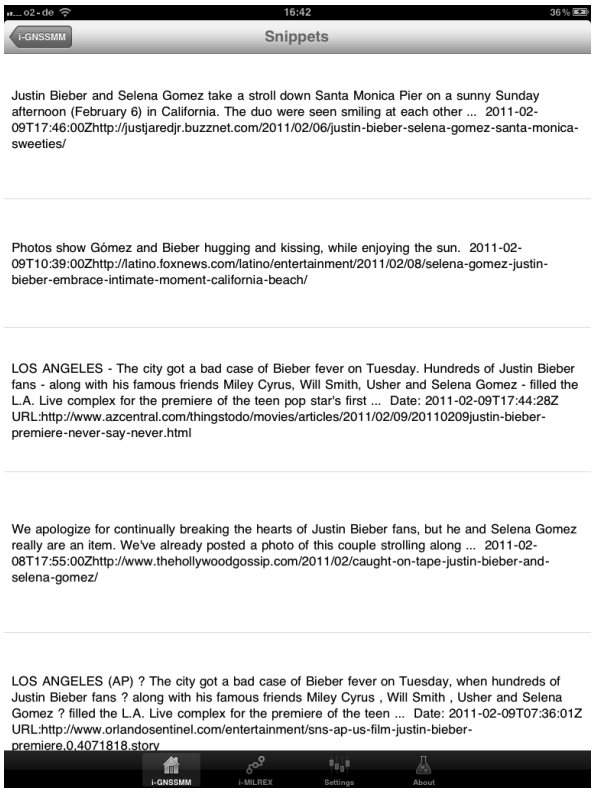

Figure 4: The snippets that are associated with the node label "selena gomez" of the topic graph from Fig. 3.In order to go back to the topic graph, the user simply touches the button labeled i-GNSSMM on the left upper corner of the iPad screen.

3 Topic Graph Extraction

We consider the extraction of a topic graph as a specific *empirical collocation extraction task*. However, instead of extracting collations between words, which is still the dominating approach in collocation extraction research, e.g., (Baroni and Evert, 2008), we are extracting collocations between chunks, i.e., word sequences. Furthermore, our measure of association strength takes into account the distance between chunks and combines it with the PMI (pointwise mutual information) approach (Turney, 2001).

The core idea is to compute a set of chunk–pair–distance elements for the N first web snippets returned by a search engine for the topic Q, and to compute the topic graph from these elements.[1] In general for two chunks, a single chunk–pair–distance element stores the distance between the chunks by counting the number of chunks in–between them. We distinguish elements which have the same words in the same order, but have different distances. For example, (Peter, Mary, 3) is different from (Peter, Mary, 5) and (Mary, Peter, 3).

We begin by creating a document S from the N-first web snippets so that each line of S contains a complete snippet. Each textline of S is then tagged with Part–of–Speech using the SVM-Tagger (Giménez and Màrquez, 2004) and chunked in the next step. The chunker recognizes two types of word chains. Each chain consists of longest matching sequences of words with the same PoS class, namely noun chains or verb chains, where an element of a noun chain belongs to one of the extended noun tags[2], and elements of a verb

[1] For the remainder of the paper N=1000. We are using Bing (`http://www.bing.com/`) for Web search.

[2] Concerning the English PoS tags, "word/PoS" expressions that match the following regular expression are considered as extended noun tag: "/(N(N|P))|/VB(N|G)|/IN|/DT". The En-

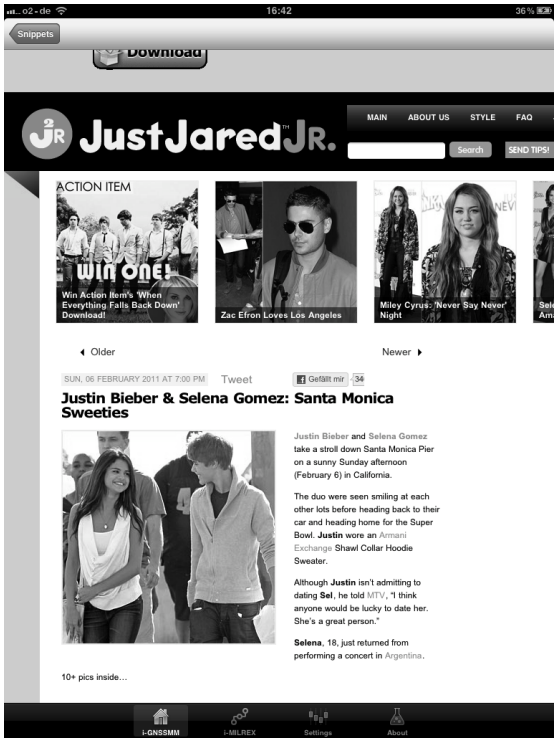

Figure 5: The web page associated with the first snippet of Fig. 4. A single touch on that snippet triggers a call to the iPad browser in order to display the corresponding web page. The left upper corner button labeled "Snippets" has to be touched in order to go back to the snippets page.

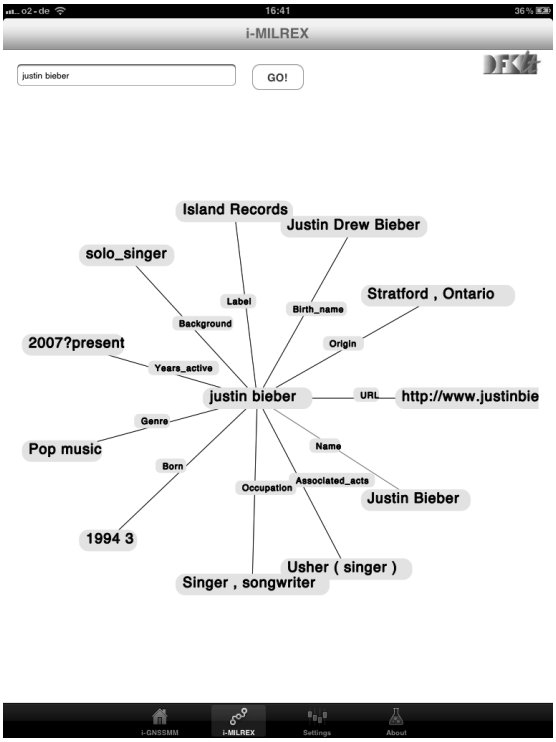

Figure 6: If mode "i–MILREX" is chosen then text exploration is performed based on relations computed from the info–boxes extracted from Wikipedia. The central node corresponds to the query. The outer nodes represent the arguments and the inner nodes the predicate of a info–box relation. The center of the graph corresponds to the search query.

chain only contains verb tags. We finally apply a kind of "phrasal head test" on each identified chunk to guarantee that the right–most element only belongs to a proper noun or verb tag. For example, the chunk "a/DT british/NNP formula/NNP one/NN racing/VBG driver/NN from/IN scotland/NNP" would be accepted as proper NP chunk, where "compelling/VBG power/NN of/IN" is not.

Performing this sort of shallow chunking is based on the assumptions: 1) noun groups can represent the arguments of a relation, a verb group the relation itself, and 2) web snippet chunking needs highly robust NL technologies. In general, chunking crucially depends on the quality of the embedded PoS–tagger. However, it is known that PoS–tagging performance of even the best taggers decreases substantially when applied on web pages (Giesbrecht and Evert, 2009). Web snippets are even harder to process because they are not necessary contiguous pieces of texts, and usually are not syntactically well-formed paragraphs due to some intentionally introduced breaks (e.g., denoted by ... betweens text fragments). On the other hand, we want to benefit from PoS tagging during chunk recognition in order to be able to identify, on the fly, a shallow phrase structure in web snippets with minimal efforts.

The chunk–pair–distance model is computed from the list of chunks. This is done by traversing the chunks from left to right. For each chunk c_i, a set is computed by considering all remaining chunks and their distance to c_i, i.e., $(c_i, c_{i+1}, dist_{i(i+1)})$, $(c_i, c_{i+2}, dist_{i(i+2)})$, etc. We do this for each chunk list computed for each web snippet. The distance $dist_{ij}$ of two chunks c_i and c_j is computed directly from the chunk list, i.e., we do not count the position

glish Verbs are those whose PoS tag start with VB. We are using the tag sets from the Penn treebank (English) and the Negra treebank (German).

of ignored words lying between two chunks.

The motivation for using chunk–pair–distance statistics is the assumption that the strength of hidden relationships between chunks can be covered by means of their collocation degree and the frequency of their relative positions in sentences extracted from web snippets; cf. (Figueroa and Neumann, 2006) who demonstrated the effectiveness of this hypothesis for web–based question answering.

Finally, we compute the frequencies of each chunk, each chunk pair, and each chunk pair distance. The set of all these frequencies establishes the chunk–pair–distance model CPD_M. It is used for constructing the topic graph in the final step. Formally, a topic graph $TG = (V, E, A)$ consists of a set V of nodes, a set E of edges, and a set A of node actions. Each node $v \in V$ represents a chunk and is labeled with the corresponding PoS–tagged word group. Node actions are used to trigger additional processing, e.g., displaying the snippets, expanding the graph etc.

The nodes and edges are computed from the chunk–pair–distance elements. Since, the number of these elements is quite large (up to several thousands), the elements are ranked according to a weighting scheme which takes into account the frequency information of the chunks and their collocations. More precisely, the weight of a chunk–pair–distance element $cpd = (c_i, c_j, D_{ij})$, with $D_{i,j} = \{(freq_1, dist_1), (freq_2, dist_2), ..., (freq_n, dist_n)\}$, is computed based on PMI as follows:

$$PMI(cpd) = log_2((p(c_i, c_j)/(p(c_i) * p(c_j)))$$
$$= log_2(p(c_i, c_j)) - log_2(p(c_i) * p(c_j))$$

where relative frequency is used for approximating the probabilities $p(c_i)$ and $p(c_j)$. For $log_2(p(c_i, c_j))$ we took the (unsigned) polynomials of the corresponding Taylor series[3] using $(freq_k, dist_k)$ in the k-th Taylor polynomial and adding them up:

$$PMI(cpd) = (\sum_{k=1}^{n} \frac{(x_k)^k}{k}) - log_2(p(c_i) * p(c_j))$$

$$, where\ x_k = \frac{freq_k}{\sum_{k=1}^{n} freq_k}$$

[3]In fact we used the polynomials of the Taylor series for $ln(1 + x)$. Note also that k is actually restricted by the number of chunks in a snippet.

The visualized topic graph TG is then computed from a subset $CPD'_M \subset CPD_M$ using the m highest ranked cpd for fixed c_i. In other words, we restrict the complexity of a TG by restricting the number of edges connected to a node.

4 Wikipedia's Infoboxes

In order to provide query specific background knowledge we make use of Wikipedia's infoboxes. These infoboxes contain facts and important relationships related to articles. We also tested DBpedia as a background source (Bizer et al., 2009). However, it turned out that currently it contains too much and redundant information. For example, the Wikipedia infobox for Justin Bieber contains eleven basic relations whereas DBpedia has fifty relations containing lots of redundancies. In our current prototype, we followed a straightforward approach for extracting infobox relations: We downloaded a snapshot of the whole English Wikipedia database (images excluded), extracted the infoboxes for all articles if available and built a Lucene Index running on our server. We ended up with 1.124.076 infoboxes representing more than 2 million different searchable titles. The average access time is about 0.5 seconds. Currently, we only support exact matches between the user's query and an infobox title in order to avoid ambiguities. We plan to extend our user interface so that the user may choose different options. Furthermore we need to find techniques to cope with undesired or redundant information (see above). This extension is not only needed for partial matches but also when opening the system to other knowledgesources like DBpedia, newsticker, stock information and more.

5 Evaluation

For an initial evaluation we had 20 testers: 7 came from our lab and 13 from non–computer science related fields. 15 persons had never used an iPad before. After a brief introduction to our system (and the iPad), the testers were asked to perform three different searches (using Google, i–GNSSMM and i–MILREX) by choosing the queries from a set of ten themes. The queries covered definition questions like *EEUU* and *NLF*, questions about persons like *Justin Bieber*, *David Beckham*, *Pete Best*, *Clark*

Kent, and *Wendy Carlos* , and general themes like *Brisbane*, *Balancity*, and *Adidas*. The task was not only to get answers on questions like "Who is . . . " or "What is . . . " but also to acquire knowledge about background facts, news, rumors (gossip) and more interesting facts that come into mind during the search. Half of the testers were asked to first use Google and then our system in order to compare the results and the usage on the mobile device. We hoped to get feedback concerning the usability of our approach compared to the well known internet search paradigm. The second half of the participants used only our system. Here our research focus was to get information on user satisfaction of the search results. After each task, both testers had to rate several statements on a Likert scale and a general questionnaire had to be filled out after completing the entire test. Table 1 and 2 show the overall result.

Table 1: Google

#Question	v.good	good	avg.	poor
results first sight	55%	40%	15%	-
query answered	71%	29%	-	-
interesting facts	33%	33%	33%	-
suprising facts	33%	-	-	66%
overall feeling	33%	50%	17%	4%

Table 2: i-GNSSMM

#Question	v.good	good	avg.	poor
results first sight	43%	38%	20%	-
query answered	65%	20%	15%	-
interesting facts	62%	24%	10%	4%
suprising facts	66%	15%	13%	6%
overall feeling	54%	28%	14%	4%

The results show that people in general prefer the result representation and accuracy in the Google style. Especially for the general themes the presentation of web snippets is more convenient and more easy to understand. However when it comes to interesting and suprising facts users enjoyed exploring the results using the topic graph. The overall feeling was in favor of our system which might also be due to the fact that it is new and somewhat more playful.

The replies to the final questions: *How successful were you from your point of view? What did you like most/least? What could be improved?* were informative and contained positive feedback. Users felt they had been successful using the system. They liked the paradigm of the explorative search on the iPad and preferred touching the graph instead of reformulating their queries. The presentation of background facts in i–MILREX was highly appreciated. However some users complained that the topic graph became confusing after expanding more than three nodes. As a result, in future versions of our system, we will automatically collapse nodes with higher distances from the node in focus. Although all of our test persons make use of standard search engines, most of them can imagine to using our system at least in combination with a search engine even on their own personal computers.

6 Acknowledgments

The presented work was partially supported by grants from the German Federal Ministry of Economics and Technology (BMWi) to the DFKI Theseus projects (FKZ: 01MQ07016) TechWatch–Ordo and Alexandria4Media.

References

Marco Baroni and Stefan Evert. 2008. *Statistical methods for corpus exploitation*. In A. Lüdeling and M. Kytö (eds.), Corpus Linguistics. An International Handbook, Mouton de Gruyter, Berlin.

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Soren Auer, Christian Becker, Richard Cyganiak, Sebastian Hellmann. 2009. *DBpedia - A crystallization point for the Web of Data*. Web Semantics: Science, Services and Agents on the World Wide Web 7 (3): 154165.

Alejandro Figueroa and Günter Neumann. 2006. *Language Independent Answer Prediction from the Web*. In proceedings of the 5th FinTAL, Finland.

Eugenie Giesbrecht and Stefan Evert. 2009. *Part-of-speech tagging - a solved task? An evaluation of PoS taggers for the Web as corpus*. In proceedings of the 5th Web as Corpus Workshop, San Sebastian, Spain.

Jesús Giménez and Lluís Màrquez. 2004. *SVMTool: A general PoS tagger generator based on Support Vector Machines*. In proceedings of LREC'04, Lisbon, Portugal.

Peter Turney. 2001. *Mining the web for synonyms: PMI-IR versus LSA on TOEFL.* In proceedings of the 12th ECML, Freiburg, Germany.