MobEx - a System for Exploratory Search on the Mobile Web

Günter Neumann and Sven Schmeier

DFKI - German Research Center for Artificial Intelligence,
Stuhlsatzenhausweg 3, 66119 Saarbrücken, Germany
{neumann,sven.schmeier}@dfki.de
http://www.dfki.de

Abstract. We present *MobEx*, a mobile touchable application for exploratory search on the mobile web. The system has been implemented for operation on a tablet computer, i.e. an Apple iPad, and on a mobile device, i.e. Apple iPhone or iPod touch. Starting from a topic issued by the user the system collects web snippets that have been determined by a standard search engine in a first step and extracts associated topics to the initial query in an unsupervised way on-demand and highly performant. This process is recursive in priciple as it furthermore determines other topics associated to the newly found ones and so forth. As a result *MobEx* creates a dense web of associated topics that is presented to the user as an interactive topic graph. We consider the extraction of topics as a specific empirical collocation extraction task where collocations are extracted between chunks combined with the cluster descriptions of an online clustering algorithm. Our measure of association strength is based on the pointwise mutual information between chunk pairs which explicitly takes their distance into account. These syntactically–oriented chunk pairs are then semantically ranked and filtered using the cluster descriptions created by a Singular Value Decomposition (SVD) approach. 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.

Keywords: Web Mining; Information Extraction; Topic Graph Exploration; Mobile Device

1 INTRODUCTION

Searching the web using standard search engines is still dominated by a passive one–tracked human-computer interaction: a user enters one or more keywords that represent the information of interest and receives a ranked list of documents. However, 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.

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 several 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 that topic and associated topics that are found.
2. The user can interactively explore this topic graph using a simple and intuitive 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.
3. Nowadays, the mobile web and mobile touchable devices, like smartphones and tablet computers, are getting more and more prominent and widespread. Thus the user might expect a device-adaptable touchable handy human–computer interaction.

In this paper, we present an approach of exploratory web search, that tackles the above mentioned requirements in the following way.

In a first step, the topic graph is computed on the fly from 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 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.

The topic graph is then displayed either on a tablet computer (in our case an iPad) as touch–sensitive graph or displayed as a stack of touchable text on a smartphone (in our case an iPhone or an iPod touch). By just selecting a node or a text box, the user can either inspect the content of a topic (i.e, the snippets or web pages) or activate the expansion of the topic graph through an on the fly computation of new related topics for the selected node. The user can request information from new topics on basis of previously extracted information by selecting a node from a newly extracted topic graph.

In such a dynamic open–domain information extraction situation, the user expects real–time performance from the underlying technology. The requested information cannot simply be pre–computed, but rather has to be determined in an unsupervised and on–demand manner relative to the current user request. This is why we assume that the relevant information can be extracted from a search engine's *web snippets* directly, and that we can avoid the costly retrieval and processing time for huge amounts of documents. Of course, direct processing of web snippets also poses certain challenges for the Natural Language Processing (NLP) components. Web snippets are usually small text summaries which are automatically created from parts of the source documents and are often only in part linguistically well–formed, cf. [Manning et al., 2008]. Thus the NLP components are required to possess a high degree of robustness and run–time behavior to process the web snippets in real–time. Since our approach should also be able to process web snippets from different languages (our current application runs for English and German), the NLP components should be easily adaptable to many languages. Finally, no restrictions to the domain of the topic should be pre–supposed, i.e., the system should be able to accept topic queries from arbitrary domains. In order to fulfill all these requirements, we are favoring and exploring the use of shallow and highly data–oriented NLP components. Note that this is not a trivial or obvious design decision, since most of the current prominent information extraction methods advocate deeper NLP components for concept and relation extraction, e.g., syntactic and semantic dependency analysis of complete sentences and the integration of rich linguistic knowledge bases like Word Net.

The paper is organized as follows. In the section 2 we briefly summarize the topic graph extraction process.[1] For the sake of completeness and readability, we present in section 3 details and examples of the user interfaces for the iPad and iPhone, respectively.

A major obstacle of the topic graph extraction process described in section 2 is its purely syntactic nature. Consequently, in section 4, we introduce a semantic clustering approach that helps to improve the quality of the extracted topics. The next sections then describe details of the evaluation of the improved topic extraction process (section 5), and present our current user experience for the iPad and iPhone user interfaces (section 6). Related work is discussed in section 7, before we conclude the paper in section 8.

2 TOPIC–DRIVEN EXPLORATION OF WEB CONTENT

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.[2] 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, (Justin, Selina, 5) is different from (Justin, Selina, 2) and (Selina, Justin, 7).

Initially, a document is created from selected web snippets so that each line contains a complete snippet. Each of these lines is then tagged with Part–of–Speech using the SVMTagger [Gimenez and Marquez., 2004] and chunked in the next step.

The chunker recognizes two types of word chains: noun chunks and verb chunks. Each recognized word chain consists of the 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 predefined extended noun tags. Elements of a verb chain only contain verb tags. For English, "word/PoS"

[1] This part of the work has partially been presented in [Neumann and Schmeier, 2011] and hence will be described and illustrated compactly.

[2] We are using Bing (http://www.bing.com/) for web search with N set to max. 1000.

expressions that match the regular expression "/(N(N|P))|/VB(N|G)|/IN|/DT" are considered as extended noun tag and for German the expression "/(N(N|E))|/VVPP|/AP|/ART". The English Verbs are those whose PoS tag start with VB (and VV in case of German). We are using the tag sets from the Penn treebank (English) and the Negra treebank (German).

The chunk–pair–distance model is computed from the list of noun group chunks.[3] This is fulfilled 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. This is to be done 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 of ignored words lying between two chunks.

Finally, we compute the chunk–pair–distance model CPD_M using the frequencies of each chunk, each chunk pair, and each chunk pair distance. CPD_M 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 $D_{ij} = \{(freq_1, dist_1), (freq_2, dist_2), ..., (freq_n, dist_n)\}$, is computed based on point–wise mutual information (PMI, cf. [Turney, 2001]) 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 using $(freq_k, dist_k)$ in the k-th Taylor polynomial and adding them up:

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

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

The visualized part of the topic graph is then computed from a subset of CPD_M using the m highest ranked chunk–pair–distance elements for fixed c_i. In other words, we restrict the complexity of a topic graph by restricting the number of edges connected to a node.

[3] The main purpose of recognizing verb chunks is to improve proper recognition of noun groups. They are ignored when building the topic graph, but see sec. 8.

3 TOUCHABLE INTERFACE FOR MOBILE DEVICES

Today, it is a standard approach to optimize the presentation of a web page, depending on the device it is displayed on, e.g., a standard or mobile web browser. Obviously, the same should hold for graphical user interfaces, and in our case, for the user interfaces designed for iPad and iPhone.

More concretely, the usage of a different mode of presentation and interaction with a topic graph depending on the device at hand, is motivated for the following reasons: For a smartphone the capabilities for displaying touchable text and graphics on one screen are limited mainly due to its relatively small screen size. Our concept for presenting the results consists of a touchable navigation based user interface which allows us to interact easily by single touch and swiping gestures. For a tablet computer with larger screens the intelligent mix of graphics and text makes a software system most appealing to the user. Hence the result presentation consists of a touchable topic graph offering multitouch capabilities like zooming and swiping.

We demonstrate our current solution by the following screenshots which show some results of the search query "Fukushima" running with the current iPad and iPhone user interfaces. In section 6 we present and discuss the outcomes of some user experiments.

3.1 Graph–based User Interface on the iPad

The screenshot in Fig. 1 shows the topic graph computed from the snippets for the query "Fukushima". 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 (using the technology described in the previous section). 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. 2. A single touch on a node which cannot be expanded automatically adds its label to the initial user query and triggers a new search with that expanded query.

Fig. 2 demonstrates how the topic graph from Fig. 1 has been expanded by a single touch on the node labeled "earthquake". Double touching on the node "fukushima dailchi" triggers the display of associated web snippets (Fig. 3) and the web pages.

3.2 Text–based User Interface on the iPhone

The next screenshots (Fig. 4 and 5) show the results of the same query displayed on the iPhone.

Fig. 4 shows the alternative representation of the topic graph displayed in Fig. 1. By single touching an item in the list the next page with associated topics to this item is shown. Finally, Fig. 5 presents the snippets after touching the item "fukushima daiichi". Touching one snippet will lead to the corresponding web page.

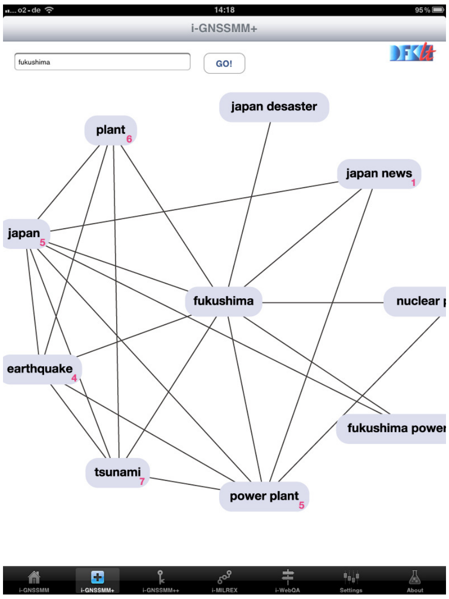


Fig. 1. The topic graph computed from the snippets for the query "Fukushima".

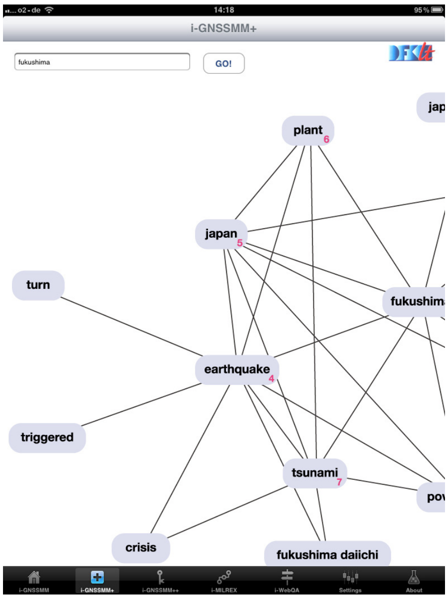
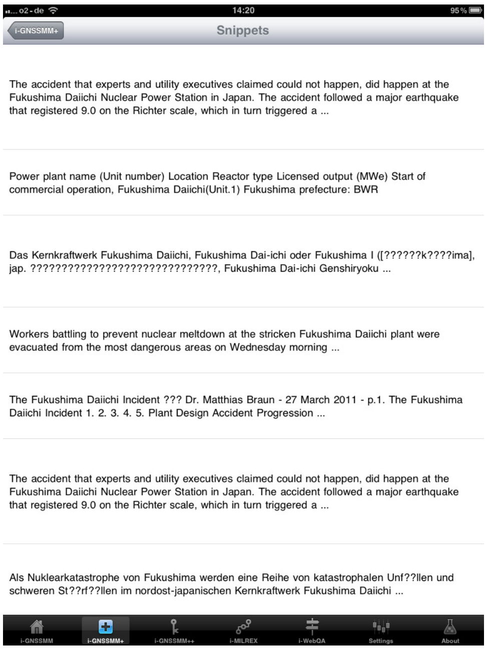


Fig. 2. The topic graph from Fig. 1 has been expanded by a single touch on the node labeled "earthquake".

4 SEMANTIC–DRIVEN FILTERING OF EXTRACTED TOPICS

The motivation for using the chunk–pair–distance statistics as described in section 2 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, and as such, are emphasizing syntactic relationships. 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 necessarily contiguous pieces of texts. For example, an initial manual analysis of a small sample revealed, that the extracted chunks sometimes are either incomplete or simply wrong. Consequently, this also caused the "readability" of the resulting topic graph due to "meaningless" relationships. Note that the decreased quality of PoS tagging is not only caused by the different style of the "snippet language", but also because PoS taggers are usually trained on linguistically more well–formed sources like newspaper articles (which is also the case for our PoS tagger in use which reports an F–measure of 97.4% on such text style).



The accident that experts and utility executives claimed could not happen, did happen at the Fukushima Daiichi Nuclear Power Station in Japan. The accident followed a major earthquake that registered 9.0 on the Richter scale, which in turn triggered a ...

Power plant name (Unit number) Location Reactor type Licensed output (MWe) Start of commercial operation, Fukushima Daiichi(Unit.1) Fukushima prefecture: BWR

Das Kernkraftwerk Fukushima Daiichi, Fukushima Dai-ichi oder Fukushima I ([??????k????ima], jap. ?????????????????????????????, Fukushima Dai-ichi Genshiryoku ...

Workers battling to prevent nuclear meltdown at the stricken Fukushima Daiichi plant were evacuated from the most dangerous areas on Wednesday morning ...

The Fukushima Daiichi Incident ??? Dr. Matthias Braun - 27 March 2011 - p.1. The Fukushima Daiichi Incident 1. 2. 3. 4. 5. Plant Design Accident Progression ...

The accident that experts and utility executives claimed could not happen, did happen at the Fukushima Daiichi Nuclear Power Station in Japan. The accident followed a major earthquake that registered 9.0 on the Richter scale, which in turn triggered a ...

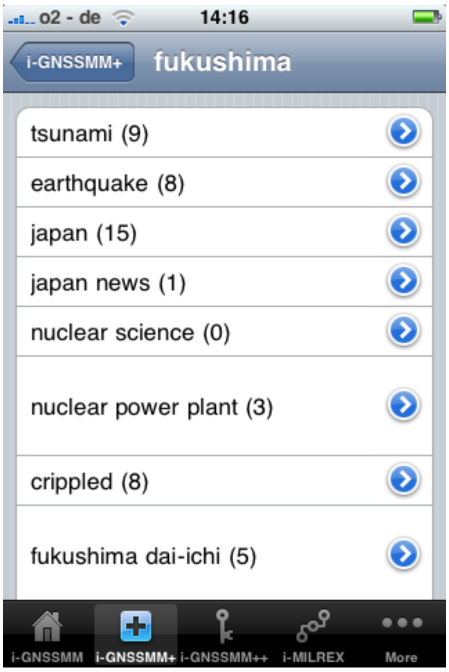
Als Nuklearkatastrophe von Fukushima werden eine Reihe von katastrophalen Unf??llen und schweren St??rf??llen im nordost-japanischen Kernkraftwerk Fukushima Daiichi ...

Fig. 3. The snippets that are associated with the node label "fukushima dai–ichi" of the topic graph from Fig. 2. A single touch on this snippet triggers a call to the iPad web browser in order to display the corresponding web page. In order to go back to the topic graph, the user simply touches the button labeled iGNSSMM on the left upper corner of the iPad screen.

Nevertheless, 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. In order to tackle this dilemma, investigations into additional semantical–based filtering seems to be a plausible way to go.

About the Performance of Chunking Web Snippets

As an initial phase into this direction we collected three different corpora of web snippets and analysed them according to the amount of well–formed sentences and incomplete sentences contained in the web snippets. Furthermore, we also randomly selected a subset of 100 snippets from each corpus and manually evaluated the quality of the PoS tagging result. The snippet corpora and results of our analysis are as follows (the shortcuts mean: #s = number of snippets retrieved, #sc = well–formed sentences within the set of snippets, #si = incomplete sentences within the snippets, #w = number of words, F(x) = F–measure achieved by the PoS tagger on a subset of 100 snippets with x words).



tsunami (9)

earthquake (8)

japan (15)

japan news (1)

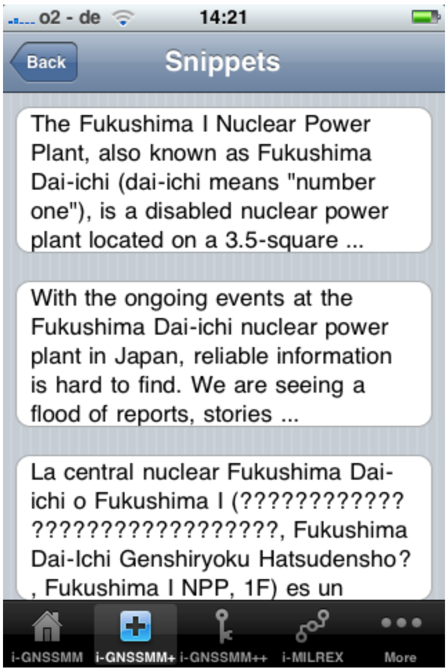
nuclear science (0)

nuclear power plant (3)

crippled (8)

fukushima dai-ichi (5)

Fig. 4. The alternative representation of the topic graph displayed in Fig. 1 on the iPhone.



The Fukushima I Nuclear Power Plant, also known as Fukushima Dai-ichi (dai-ichi means "number one"), is a disabled nuclear power plant located on a 3.5-square ...

With the ongoing events at the Fukushima Dai-ichi nuclear power plant in Japan, reliable information is hard to find. We are seeing a flood of reports, stories ...

La central nuclear Fukushima Dai-ichi o Fukushima I (???????????? ?????????????????, Fukushima Dai-Ichi Genshiryoku Hatsudensho? , Fukushima I NPP, 1F) es un

Fig. 5. The snippets after touching the item “fukushima daiichi”.

Fukushima This corpus represents snippets mainly coming from official online news magazines. The corpus statistics are as follows:

#s	#sc	#si	#w	F(2956)
240	195	182	6770	93.20%

Justin Bieber This corpus represents snippets coming from celebrity magazines or gossip forums. The corpus statistics are:

#s	#sc	#si	#w	F(3208)
240	250	160	6420	92.08%

New York This corpus represents snippets coming from different official and private homepages, as well as from news magazines. The corpus statistics are:

#s	#sc	#si	#w	F(3405)
239	318	129	6441	92.39%

This means that 39% of all tagged sentences have been incomplete and that the performance of the Pos tagger decreased by about 5% F–measure (compared to the reported 97.4% on newspaper). Consequently, a number of chunks are incorrectly recognized. For example, it turned out that date expressions are systematically tagged as nouns, so that they will be covered by our noun chunk recognizer although they should not (cf. section 2). Furthermore, the genitive possessive (the "'s" as in "Japan's president") was classified wrongly in a systematic way which also had a negative effect on the performance of the noun chunker. Very often nouns were incorrectly tagged as verbs because of erroneously identified punctuation. Thus, we need a filtering mechanism that is able to identify and remove the wrongly chunked topic–pairs.

Semantic Filtering of Noisy Chunk Pairs

A promising algorithmic solution to this problem is provided by the online clustering system *Carrot2* [Osinski and Weiss, 2008] that computes sensible descriptions of clustered search results (i.e., web documents). The Carrot2 system is based on the Lingo [Osinski et al., 2004] algorithm. Most algorithms for clustering open text follow a kind of "document–comes–first" strategy, where the input documents are clustered first and then, based on these clusters, the descriptive terms or labels of the clusters are determined, cf. [Geraci et al., 2006]. The Lingo algorithm actually reverses this strategy by following a three–step "description–comes–first" strategy (cf. [Osinski et al., 2004] for more details): 1) extraction of frequent terms from the input documents, 2) performing reduction of the (pre–computed) term–document matrix using Singular Value Decomposition (SVD) for the identification of latent structure in the search results, and 3) assignment of relevant documents to the identified labels.

The specific strategy behind the Lingo algorithm matches our needs for finding meaningful semantic filters very well: we basically use step 1) and 2) to compute a set of meaningful labels from the web snippets determined by a standard search engine as described in section 2. According to the underlying latent semantic analysis performed by the Lingo algorithm, we interpret the labels as semantic labels. We then use these labels and match them against the ordered list of chunk–pair–distance elements computed in the topic extraction step described in section 2. This means that all chunk–pair–distance elements that do not have any match with one of the semantic labels are deleted.

The idea is that this filter identifies a semantic relatedness between the labels and the syntactically determined chunks. Since we consider the labels as semantic topics or classes, we assume that the non-filtered pairs correspond to topic–related (via the user query) relevant relationships between semantically related decriptive terms.

Of course, it actually remains to evaluate the quality and usefullness of the extracted topics and topic graph. In the next sections we will discuss two directions: a) a quantitative evaluation against the recognition of different algorithms for identifying named entities and other rigid identifiers, and b) a qualitative evaluation by means of the analysis of user experience.

5 EVALUATION OF THE EXTRACTED TOPICS

Our topic extraction process is completely unsupervised and web–based, so evaluation against standard gold corpora is not possible, because they simply do not yet exist (or at least, we do not know about them). For that reason we decided to compare the outcome of our topic extraction process with the outcomes of a number of different recognizers for named entities (NEs).

Note that very often the extracted topics correspond to rigid designators or generalized named entities, i.e., instances of proper names (persons, locations, etc.), as well as instances of more fine grained subcategories, such as museum, river, airport, product, event (cf. [Nadeau and Sekine, 2007]). So seen, our topic extraction process (abbreviated as *TEP*) can also be considered as a query–driven context–oriented named entity extraction process with the notable restriction that the recognized entities are unclassified. If this perspective makes sense, then it seems plausible to measure the degree of overlap between our topic extraction process and the recognized set of entities of other named entity components to learn about the coverage and quality of *TEP*.

For the evaluation of *TEP* we compared it to the results of four different NE recognizers:

1. SProUT[Drozdzynski et al., 2004]: The *SProUT*–system is a shallow linguistic processor that comes with a rule–based approach for named entity recognition.
2. AlchemyAPI[4]: *AlchemyAPI*–system uses statistical NLP and machine learning algorithms for performing the NE recognition task.
3. Stanford NER[Dingare et al., 2004]: The *Stanford NER*–system uses a character based Maximum Entropy Markov model trained on annotated corpora for extracting NEs.
4. OpenNLP[5]: A collection of natural language processing tools which use the Maxent package to resolve ambiguity, in particular for NE recognition.

We tested all systems with the three snippet corpora described in section 4.

The tables 1, 2, and 3 show the main results for the three different corpora; table 4 shows the results summarised. All numbers denote percentages that show how many relevant[6] NEs of the algorithm in the row could be extracted by the algorithm in the column. For example, in the dataset “Justin Bieber” *TEP* extracted 85.37% of the NEs which have been extracted by *SProUT*. *AlchemyAPI* extracted 75.64% and *StanfordNER* extracted 78.95% of the NEs that have been extracted by *SProUT*. The numbers with preceding “#” show the number of extracted NEs. The following roman numbers are used to denote the different algorithms: I=*SProUT*, II=*AlchemyAPI*, III=*StanfordNER*, IV=*OpenNLP*, and V=*TEP*.

[4] http://www.AlchemyAPI.com

[5] http://incubator.apache.org/opennlp/

[6] Relevance here means that a NE must occur more than 4 times in the whole dataset. The value has been experimentally determined.

Table 1. Results for query *Justin Bieber.*

	I	II	III	IV	V
I	#136	75.64	78.95	78.48	85.37
II	69.01	#143	93.97	86.00	97.17
III	76.71	97.52	#172	92.86	96.09
IV	74.70	89.19	88.52	#196	95.10
V	67.77	79.61	80.66	81.13	#157

Table 2. Results for query *Fukushima.*

	I	II	III	IV	V
I	#121	81.03	83.61	81.35	87.5
II	80.26	#129	93.46	87.36	98.48
III	85.00	94.59	#131	91.67	92.22
IV	74.65	89.13	85.26	#178	91.58
V	72.93	80.04	83.19	82.26	#132

Table 3. Results for query *New York.*

	I	II	III	IV	V
I	#175	81.39	88.24	85.15	71.05
II	76.60	#169	93.53	86.51	74.36
III	90.00	95.79	#280	92.35	73.28
IV	84.43	92.72	93.17	#230	83.49
V	81.11	83.90	73.77	79.87	#166

Table 4. Summary for NER Evaluation.

	I	II	III	IV	V
I	#432	79,25	83.6	81.66	81.31
II	75.29	#441	93.65	86.62	90.00
III	83.90	95.97	#583	92.29	87.19
IV	83.90	95.97	583	#604	87.19
V	73.94	81.18	79.21	81.09	#455

Keeping in mind that our approach always starts with a topic around which all the NEs are grouped, i.e. NE recognition is biased or directed, it is hard to define a gold standard, i.e. manually annotate all NEs which are important in a specific context. In context of the query "Fukushima" most people would agree that word groups describing the nuclear power plant disaster clearly are NEs. Some would also agree that terms like "earthquake" or "tsunami" function as NEs too in this specific context. Given a query like "New York" people probably would not agree that "earthquake" should function as a specific term in this context. Of course there are NEs of generic type like "persons", "locations", or "companies", but it is questionable whether they suffice in the context of our task.

Hence we compared the systems directly with the results they computed. The main interest in our evaluation was whether the extracted NEs by one algorithm can also be extracted by the other algorithms. Furthermore, we set a very simple rating scheme telling us that detected NEs with more occurences are more important than those with lower frequencies.[7]

The results show that, looking at the numbers and percentages, no system outperforms the others, which on the other hand confirms our approach. Please note that the *TEP* approach works for query-driven context-oriented named entity recognition only. This means that all approaches used in this evaluation clearly have their benefits in other application areas.

Nevertheless by going into details we saw some remarkable differences between the results the systems produced. All systems were able to extract the main general NEs like locations or persons. For terms that are important in the context of actuality and current developments, we saw that the *TEP* approach is able to extract more relevant items. In case of "Fukushima", the *SProUT* system did not extract terms like "eartquake", "tsunami" or "nuclear power plant". Of

[7] Except for the *TEP*, where we used the PMI as described above.

course this is because the underlying ruleset has not been developed for covering such types of terms. The *AlchemyAPI* and *StanfordNER* systems were able to extract these terms but failed in detecting terms like "accident" or"safety issues". For "Justin Bieber" relevant items like "movie", "tourdates" or "girlfriend" could not be detected by all systems except *TEP* . For the snippets associated with the query "New York" all systems identified the most important NEs, and differed for less important NEs only.

Last but not least the runtime, which plays an important role in our system, varied from 0.5 seconds for the *SProUT* system, to 2 seconds for *TEP*, 4 seconds for *StanfordNER* to 15 seconds for *AlchemyAPI.*

6 EVALUATION OF THE TOUCHABLE USER INTERFACE

For information about the user experience we had 26 testers — 20 for testing the iPad App and 6 for testing the iPhone App: 8 came from our lab and 18 from non–computer science related fields. 15 persons had never used an iPad before, 4 persons have been unfamiliar with smartphones. More than 80 searches have been made with our system and with Google respectively.

After a brief introduction to our system (and the mobile devices), the testers were asked to perform three different searches (using our system on the iPad, iPhone and Google on the iPad/iPhone) 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 iPad–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 iPad–testers used only our system. Here our research focus was to get information on user satisfaction of the search results. The iPhone–testers always used Google and our system mainly because they were fewer people.

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. The tables 5, 6, 7, and 8 show the overall result.

The results show that people prefer the result representation and accuracy in the Google style when using the iPad. Especially for the general themes the presentation of web snippets is more convenient and easier to understand. The iPhone–testers could be divided into two groups: in case they were unfamiliar with smartphones the testers preferred our system because it needs much less user interaction and the result are more readable. Testers being familiar with smartphones again prefered the Google style mainly because they are used to it.

#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%

Table 5. System on the iPad

#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 6. Google on the iPad

#Question	v.good	good	avg.	poor
results first sight	31%	46%	23%	-
query answered	70%	20%	10%	-
interesting facts	45%	36%	19%	-
suprising facts	56%	22%	11%	11%
overall feeling	25%	67%	8%	-

Table 7. System on the iPhone

#Question	v.good	good	avg.	poor
results first sight	23%	63%	7%	7%
query answered	70%	20%	10%	-
interesting facts	33%	33%	33%	-
suprising facts	36%	-	27%	37%
overall feeling	25%	33%	33%	9%

Table 8. Google on the iPhone

However, when it comes to interesting and suprising facts users enjoyed exploring the results using the topic graph (iPad) or the navigation based representation (iPhone/iPod). 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. For the iPhone they prefered the result representation in our system in general and there have been useful comments for improving it. One main issue is the need of a summary or a more knowledge based answer to the search query as Google often does it by offering a direct link to wikipedia as a first search result. This will be part of our future research.

Although all of our test persons make use of standard search engines, most of them can imagine to use our system at least in combination with a search engine on their own mobile devices. The iPhone test group even would use our system as their main search tool (on the smartphone) when the proposed improvements have been implemented.

7 RELATED WORK

Our approach is unique in the sense that it combines interactive topic graph extraction and exploration on different mobile devices with recently developed technology from exploratory search, text mining and information extraction methods. As such, it learns from and shares ideas with other research results. The most relevant ones are briefly discussed below.

Exploratory Search [Marchionini, 2006] distinguishes three types of search activities: a) lookup search, b) searching to learn, and c) investigative search, where b) and c) are considered as forms of exploratory search activities. Lookup search corresponds to fact retrieval, where the goal is to find precise results for carefully specified questions with minimal need for examinating and validating the result set. The learn search activity can be found in situations where the found material is used to develop new knowledge and basically involves multiple iterations of search. It is assumed that the returned set of objects maybe instantiated in various media, e.g., graphs, maps or texts. Investigative searching is a next level of search activity that supports investigation into a specific topic of interest. It also involves multiple iterations even for very long periods and the results are usually strictly assessed before they are integrated into knowledge bases. Our proposed approach of exploratory search belongs to the searching to learn activity. In this spirit, our approach is more concerned with recall (maximizing the number of possibly relevant associated topics that are determined) than precision (minimizing the number of possibly irrelevant associated topics that are determined).

Collocation Extraction We consider the extraction of a topic graph as a specific *empirical collocation extraction task*. However, instead of extracting collocations 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].

[Geraci et al., 2006] also exploit the benefit of Web snippets for improved internet search by grouping the web snippets returned by auxiliary search engines into disjoint labeled clusters. As we do, they also consider methods for automatic labeling. However, their focus is on improving clustering of terms and not on the extraction of empirical collocations between individual terms. Furthermore, they advocate the "document–comes–first" approach of clustering Web snippets which is inappropriate for our methodology, cf. sec. 4.

Unsupervised Information Extraction Web–based approaches to unsupervised information extraction have been developed by Oren Etzioni and colleagues, cf. [Banko et al., 2007]; [Etzioni, 2007]; [Yates, 2007]. They developed a range of systems (e.g., KnowItAll, Textrunner, Resolver) aimed at extracting large collections of facts (e.g., names of scientists or politicians) from the Web in an unsupervised, domain-independent, and scalable manner. They also argue for light–weight NLP technologies and follow a similar approach to chunk extraction as we do (but not a chunk–pair–distance statistics). Although we do not yet explicitly extract relations in the sense of standard relation extraction, our topic graph extraction process together with the clustering mechanism can be extended to also support relation extraction, which will be a focus of our next research.

8 CONCLUSION AND OUTLOOK

We presented an approach of interactive topic graph extraction for exploration of web content. The initial information request is issued online by a user to the system in the form of a query topic description. The topic query is used for constructing an initial topic graph from a set of web snippets returned by a standard search engine. At this point, the topic graph already displays a graph of strongly correlated relevant entities and terms. The user can then request further detailed information through multiple iterations.

A prototype of the system has been realized on the basis of two specialized mobile touchable user interfaces for operation on an iPad and on an iPhone which receive both the same topic graph data structure as input. We believe that our approach of interactive topic graph extraction and exploration, together with its implementation on a mobile device, helps users explore and find new interesting information on topics about which they have only a vague idea or even no idea at all.

Our next future work will consider the integration of open shared knowledge bases into the learn search activity, e.g., Wikipedia or other similar open web knowledge sources and the extraction of relations, and finally to merge information from these different resources. We already have embedded Wikipedia's infoboxes as background knowledge but not yet integrated them into the extracted web topic graphs, cf. [Neumann and Schmeier, 2011] for some more details. If so done, we will investigate the role of Wikipedia and the like as a basis for performing disambiguation of the topic graphs. For example, currently, we cannot distinguish the associated topics extracted for a query like "Jim Clark" whether they are about the famous formula one racer or the Netscape founder or even about another person.

In this context, the extraction of semantic relations will be important. Currently, the extracted topic pairs only express certain semantic relatedness, but the nature and meaning of the underlying relationship is unclear. We have begun investigating this problem by extending our approach of chunk–pair–distance extraction to the extraction of triples of chunks with already promising initial results.

ACKNOWLEDGEMENTS

The presented work was partially supported by grants from the German Federal Ministry of Economics and Technology (BMWi) to the DFKI THESEUS project (FKZ: 01MQ07016).

References

[Banko et al., 2007] Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open information extraction from the web. In *Proceedings of IJCAI–2007, pp 2670–2676.*

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

[Dingare et al., 2004] Dingare, S., Nissim, M., Finkel, J., Grover, C., and Manning, C. D. (2004). A system for identifying named entities in biomedical text: How results from two evaluations reflect on both the system and the evaluations. In *Comparative and Functional Genomics 6:pp 77-85.*

[Drozdzynski et al., 2004] Drozdzynski, W., Krieger, H.-U., Piskorski, J., Schäfer, U., and Xu, F. (2004). Shallow processing with unification and typed feature structures — foundations and applications. *Künstliche Intelligenz*, pages 17–23.

[Etzioni, 2007] Etzioni, O. (2007). Machine reading of web text. In *Proceedings of the 4th international Conference on Knowledge Capture, Whistler, BC, Canada, pp 1-4.*

[Geraci et al., 2006] Geraci, F., Pellegrini, M., Maggini, M., and Sebastiani, F. (2006). Cluster generation and labeling for web snippets: A fast, accurate hierarchical solution. *Journal of Internet Mathematics*, 4(4):413–443.

[Giesbrecht and Evert, 2009] Giesbrecht, E. and Evert, S. (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.*

[Gimenez and Marquez., 2004] Gimenez, J. and Marquez., L. (2004). Svmtool: A general pos tagger generator based on support vector machines. In *Proceedings of LREC'04, pp. 43 - 46.*

[Manning et al., 2008] Manning, C. D., Raghavan, P., and Schütze, H. (2008). Introduction to information retrieval. In *Cambridge University Press.*

[Marchionini, 2006] Marchionini, G. (2006). Exploratory search: from finding to understanding. *Commun. ACM*, 49(4):41–46.

[Nadeau and Sekine, 2007] Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Journal of Linguisticae Investigationes*, 30(1):1–20.

[Neumann and Schmeier, 2011] Neumann, G. and Schmeier, S. (2011). A mobile touchable application for online topic graph extraction and exploration of web content. In *Proceedings of the ACL-HLT 2011 System Demonstrations.*

[Osinski et al., 2004] Osinski, S., Stefanowski, J., and Weiss, D. (2004). Lingo: Search results clustering algorithm based on singular value decomposition. In *Proceedings of the International IIS: Intelligent Information Processing and Web Mining Conference. Advances in Soft Computing, Springer.*

[Osinski and Weiss, 2008] Osinski, S. and Weiss, D. (2008). Carrot2: Making sense of the haystack. In *ERCIM News.*

[Turney, 2001] Turney, P. (2001). Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of ECML–2002. Freiburg, Germany, pp 491-502.*

[Yates, 2007] Yates, A. (2007). Information extraction from the web: Techniques and applications. In *Ph.D. Thesis, University of Washington, Computer Science and Engineering.*