Visualization of news articles

Marko Grobelnik, Dunja Mladenić

Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
{marko.grobelnik, dunja.mladenic}@ijs.si

Keywords: text visualization, context visualization, document clustering, large collection of news stories

Received: October.2004

This paper presents a system for visualization of large amounts of new stories. In the first phase, the new stories are preprocessed for the purpose of name-entity extraction. Next, a graph of relationships between the extracted name entities is created, where each name entity represents one vertex in the graph and two name entities are connected if they appear in the same document. The graph of entities is presented as a local neighborhood enriched with additional contextual information in the form of characteristic keywords and related name entities connected to the entity in the focus. Operations for browsing a graph are implemented to be efficient enabling quick capturing of large amounts of information present in the original text.

1 Introduction

Text visualization is an area having the main goal to present textual contents of one or many documents in a visual form. The intention of producing visualization of the textual contents is mainly to create graphical form of the content summary on different levels of abstraction.

In general, we can say that ideas used in text visualization algorithms come primarily from data analysis research areas (such as statistics, machine learning, data mining) [1, 2, 3, 4] where data visualization play important role as a key technique for showing the data and results of analytic methods. Textual data is in this respect just another type of data with its specific properties which need to be taken into account when visualizing it. Main characteristics relevant for text visualization are [5]:

- High data dimensionality when using typical bag-of-words representation, where each word and each phrase represents one dimension in the data space.
- High redundancy, meaning that many dimensions can be easily merged into one dimension without loosing much information. This is caused by the two properties of words, namely synonymy (different surface word forms having the same meaning – e.g. singer, vocalist) and hyponymy (one word denotes a subclass of an another – e.g. breakfast, is a subclass of a meal)
- Ambiguity between words in the cases where the same surface form of the word has different meanings (homonomy – e.g. the word 'bank' can mean 'river bank' or 'financial institution') or in the cases where the name form has related meaning (polysemy – e.g. 'bank' can mean 'blood bank' or 'financial institution')
- Frequency of words (and phrases) follows power distribution, meaning that we deal with small number of very frequent words and high number of infrequent words. Having this in mind, we need to use appropriate weighting schemas (e.g., most popular being TFIDF) to normalize importance of the words to be able to work with the standard data analytic techniques.

Furthermore, when talking about text visualization we also need to be aware of the type of text we are dealing with. Namely, different document types have different characteristics which need to be considered when designing an efficient text visualization mechanism. Some examples of such different types of textual data are: Web documents (being typically short, having linkage structure and additional formatting information), e-mails and news-group postings (short documents with specific internal structure, appearing in content threads and using specific language), customer reports, chat rooms discussions, literature, legal documents, technical text, news stories etc.

In this paper we are dealing with news stories. Specifically, we have designed and developed a system for preprocessing and visualizing large amounts of documents coming from a news wire. In general, news stories are special type of text having most often the following properties:

- short documents,
- written by professionals,
- low number of language mistakes,
- having good rhetorical structure,
- rich information about people, companies, places, etc.,
- a single news document containing pieces of larger stories usually spanning over several documents.

Our approach takes into account the above properties giving a special emphasis on the last two items namely, named objects (such as people, companies, and places) and the context they are appearing in.

In the following sections we present related work, sample news articles corpus, design and architecture of the system, name entity extraction, keyword extraction, browsing and visualization user interface and discussion at the end.

2 Related work

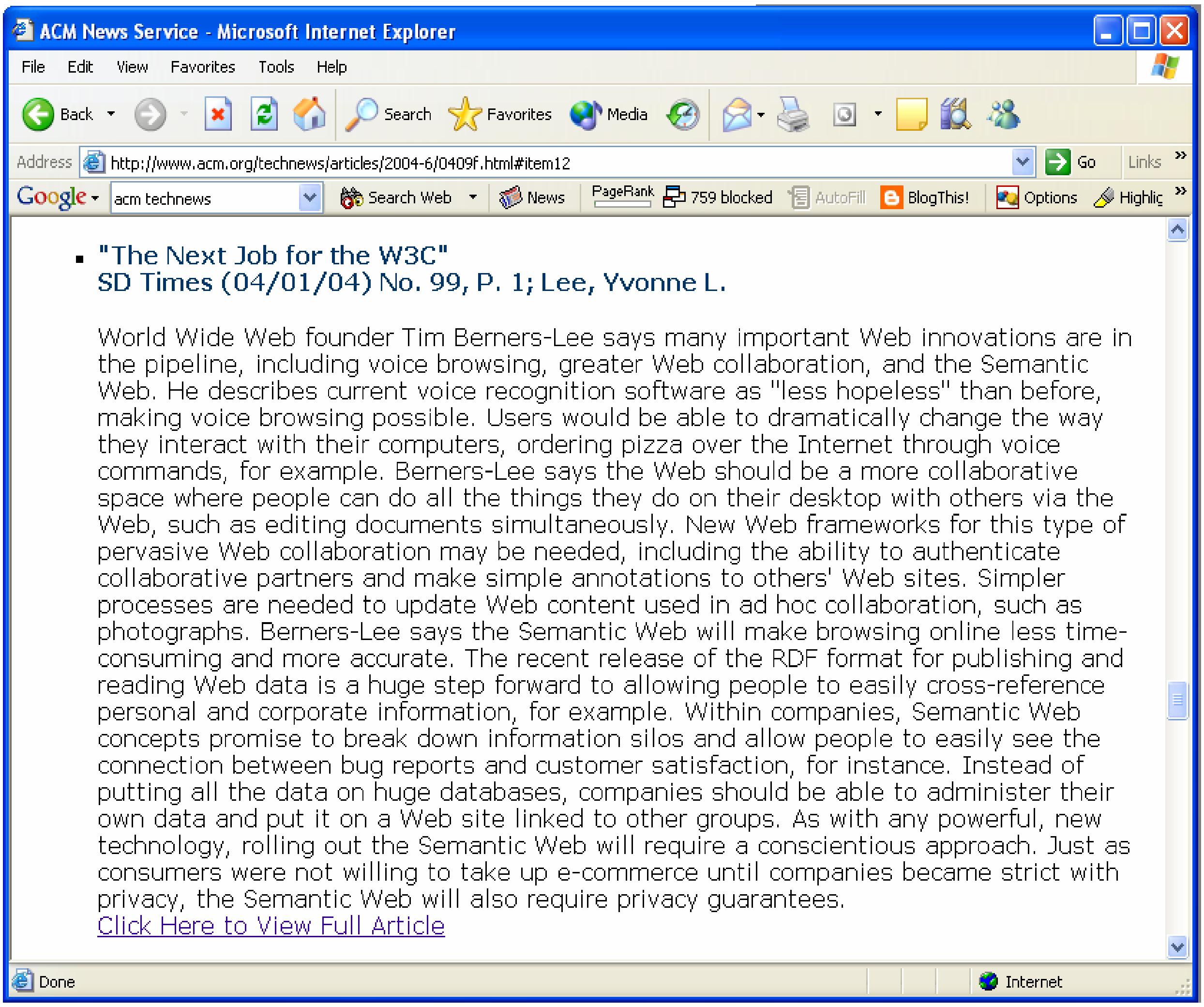
Wider area of the work presented in this paper is data visualization [3] and in particular text visualization [6]. This work also fits in the recent developments of semantic web in particular visualization of ontologies and other knowledge structures [1].

In this paper we are dealing with visualization and browsing of news stories which require special treatment. In the literature there are not many published works on this specific subtopic. Most prominent is the overview publication from MITRE team [7] giving good overview over the approaches for visualization of different document types, including news stories. Their goals are similar to the work presented here, but the actual approach is quite different. Their publication appeared also at [8] together with some other interesting approaches for document visualization.

Another approach for visualizing trends in news documents is the system ThemeRiver [9] developed at Pacific Northwest National Laboratory together with many other interesting approaches for information text visualization [10]. ThemeRiver in particular is specialized for analyzing and visualizing trends in news stories over time, enabling efficient detection of trends in the vocabulary used in the texts. Among others, we would also like to mention our previous work on visualization of large text corpora [11].

3 Sample news corpus

The functionality of our approach is presented here on a corpus of news articles from “ACM Technology News” service at http://www.acm.org/technews/archives.html. The corpus includes general news from the most areas of Information Technology (from December 1999 on). It includes over 11.000 article summaries of the length 200-400 words. Figure 1 shows a typical article summary from the corpus which is used in the subsequent procedure.


ACM News Service - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
http://www.acm.org/technews/articles/2004-6/0409f.html#item12
Go
Links
Google
acm technews
Search Web
News
PageRank
759 blocked
AutoFill
BlogThis!
Options
Highlic
Done
Internet

- "The Next Job for the W3C"
  SD Times (04/01/04) No. 99, P. 1; Lee, Yvonne L.

  World Wide Web founder Tim Berners-Lee says many important Web innovations are in the pipeline, including voice browsing, greater Web collaboration, and the Semantic Web. He describes current voice recognition software as "less hopeless" than before, making voice browsing possible. Users would be able to dramatically change the way they interact with their computers, ordering pizza over the Internet through voice commands, for example. Berners-Lee says the Web should be a more collaborative space where people can do all the things they do on their desktop with others via the Web, such as editing documents simultaneously. New Web frameworks for this type of pervasive Web collaboration may be needed, including the ability to authenticate collaborative partners and make simple annotations to others' Web sites. Simpler processes are needed to update Web content used in ad hoc collaboration, such as photographs. Berners-Lee says the Semantic Web will make browsing online less time-consuming and more accurate. The recent release of the RDF format for publishing and reading Web data is a huge step forward to allowing people to easily cross-reference personal and corporate information, for example. Within companies, Semantic Web concepts promise to break down information silos and allow people to easily see the connection between bug reports and customer satisfaction, for instance. Instead of putting all the data on huge databases, companies should be able to administer their own data and put it on a Web site linked to other groups. As with any powerful, new technology, rolling out the Semantic Web will require a conscientious approach. Just as consumers were not willing to take up e-commerce until companies became strict with privacy, the Semantic Web will also require privacy guarantees.
  Click Here to View Full Article

Figure 1. Example of a news article summary from ACM-TechNew

4 Design and architecture of the system

The main goal, when designing the system called "Contexter", was to help expert and semi-expert users (such as analysts, journalists, social scientists, experienced web surfers) to get an efficient and quick understanding of large corpus of general news stories providing different levels of abstraction. This is to be achieved by several means:

- by showing relationships between entities appearing within documents,
- by calculating and showing contexts within which the entities appear either individually or in combination with other entities,
- by using several types of visualization simultaneously,
- by efficient and responsive graphical user interface enabling easy moving from abstract to detailed information.

One of the fundamental design assumptions is that most of the relevant information is centered around the entities mentioned within the documents. In our context, entities can be names of people, names of companies and other institutions, geographical names and places, product names, etc. An additional property of entities is that they serve as connectors between different documents forming longer threads of stories which are not explicitly noted with typical news corpora (usually such information is not present in meta-data of news articles). Based on these observations, our basic representation of documents within the news corpus is three-fold: (1) plain text as originally provided, (2) bag-of-words representation of the text, (3) representation by a set of name-entities.

1. **Plain news text as written by the authors**.

This representation is used exclusively for showing the document content to the user, when the user comes to the point that s/he explicitly requests the full textual information. This representation offers lowest level of content abstraction.

2. **Bag-of-words representation using some kind of weighting schema** (in our case TFIDF).

In this case we still include most of the words appearing in the original text – we just delete the stop-words (non-informative functional words), perform stemming (unifying different surface forms for the same words), pre-calculate phrases (frequent and significant consecutive sequences of several words), and the most important, ignore the order of the words (for the purpose of more efficient computation). The goal of this representation is to efficiently calculate contexts in the form of keyword lists to allow for a higher abstraction of the contents compared to the plain text.

3. **Set of name-entities appearing within the article**.

In our case we use variant of relatively standard name-entity extraction algorithm based on word capitalization (primary candidates for the name-entities are the words starting with capital letter) with additional mechanism for name consolidation (detecting that e.g. 'Bill Clinton'=='President Clinton'=="Clinton'). This representation in our case offers the highest abstraction level for an individual document. Because of its structured nature (e.g. names are consolidated on the level of the whole news corpus) it serves as a connecting level between different documents.

On the input to the system we get a set of documents representing news articles. We have no special assumptions on the form, structure and meta-data within the documents – main element is textual part of the documents which is further processed. Next, the documents are preprocessed in two different ways. First, the text is cleaned and the bag-of-words representation is created, and next, the name-entities are extracted. All the documents are stored in the database in three different representations (as already described: plain text, bag-of-words and name-entities). The database is used by the client software using efficient graphical user interface described in the following sections. Figure 2 shows the architecture of the system.

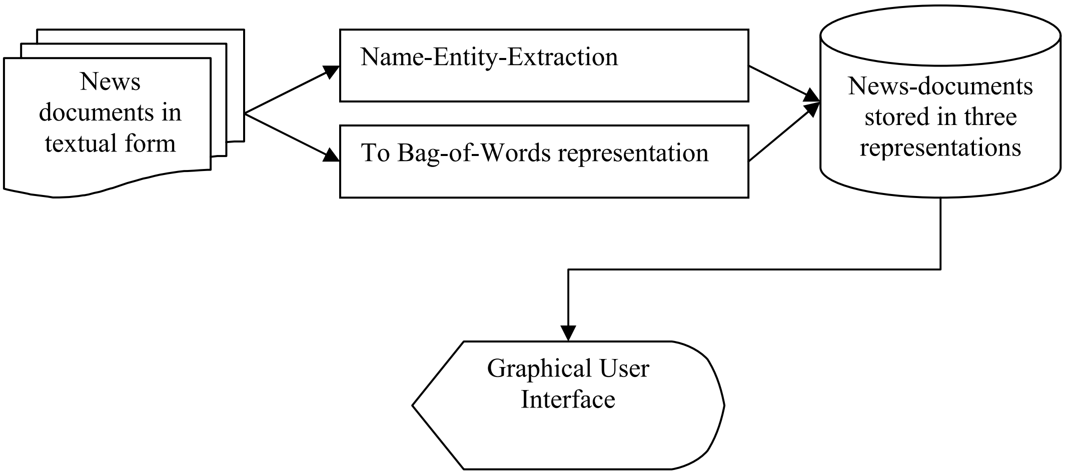

News documents in textual form
Name-Entity-Extraction
To Bag-of-Words representation
News-documents stored in three representations
Graphical User Interface

Figure 2. Architecture of the "Contexter" system

4.1 Named entity extraction

Information extraction and specifically name-entity-extraction [4] are one of the most popular areas of text mining. The main function is to convert parts of unstructured textual data into structured form which enables to use standard data analytic methods available in data mining and statistical packages (e.g. SAS and SPSS use this kind of approach). There are three main approaches when extracting useful pieces of information from text: (1) manual extraction rules, (2) automatically generated rules with machine learning methods, and (3) hybrid methods combining the two approaches. In everyday practice the approach with manual rules seems to be the most effective and frequently used. While machine learning methods give good results on datasets with lack of domain knowledge, the automatically generated rules usually need human corrections and additions to be practically useful. In general, for controlled corpora, initial investment needed to get good results with manually modified or even created rules seems to be the most price-performance effective.

In our case, name entity extraction algorithm is based on one of the most typical heuristic approaches – on word capitalization. This approach usually gives good results on high quality texts and introduces low overhead in terms of computational efficiency and additional tuning of parameters. Furthermore, it gives good results for most of the western languages without any special tuning (except for German which uses capital letters for all the nouns). Main characteristic of the method is that it provides very good recall (almost all of the real name-entities are proclaimed as name-entities), but slightly lower precision (some of the proclaimed name-entities are not name-entities) which is in practical setting enough – errors and exceptions are handled separately by the list of exceptions. Empirical evaluation of the method (based on 100 randomly selected news articles) showed precision value of 73% and recall value of 96% – recall and precision are standard Information Retrieval evaluation measures measuring the ‘truth’ and the ‘whole truth’ of the result set.

In addition to the name-entity extraction, we also use name consolidation mechanism which tries to unify different surface forms into one name-entity (e.g. ‘Bill Clinton’==’President Clinton’==”Clinton’). For this purpose we use heuristic approach based on the phrase similarity.

4.2 Bag-of-Words representation and Keyword Extraction

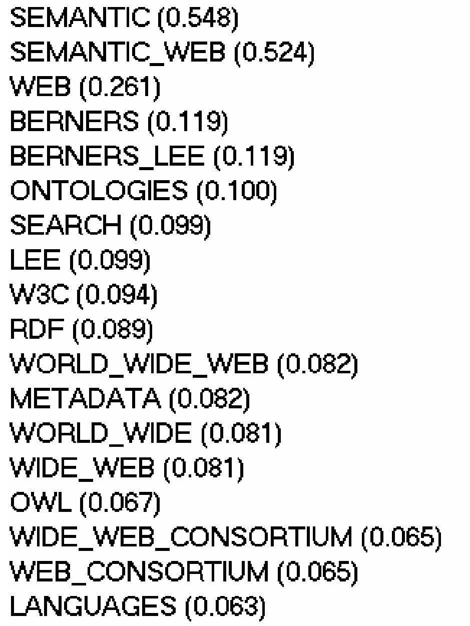
Classical representation of documents in Information Retrieval is so called the bag-of-words (or word-vector) representation [2, 4, 5]. It enables efficient execution of several fundamental operations on the transformed text documents. The idea of bag-of-words representation is to represent each document as a vector of numeric variables, where each variable represents one word (or phrase) from the dictionary (union of all words from all the documents in the corpus). If a particular words appear within a document, then its vector includes non-zero value for the word-variable (usually number of appearances of the word within the document), otherwise, the value is zero. Since most of the values within the single document vector are zero, this calls for more efficient representation of the vector – typically vectors are represented with so called “sparse vector representation” which is an ordered set of pairs *(WordId, Weight)*, where *WordId* denotes word and *Weight* non-zero frequency of the word within the document (usually called term-frequency).

An important issue when dealing with bag-of-words representation is how to represent the word weights. Using plain term-frequency is usually not enough, because the power-distribution of the words (small number of very frequent words and high number of infrequent words) damages performance of most of the analytic methods. Therefore, we use one of the improved heuristic weighting schemas which correct the influence of the word distributions. The most popular weighting schema is TFIDF which calculates a weight for each word within each document using the following formula:

$$tfidf(w) = tf \,.\, \log(\frac{N}{df(w)})$$

In the above formula *tf* stands for the term-frequency (the number of word appearances within a document), *df* stands for the document frequency (the number of documents in which the words appears), and *N* is the number of all documents within the corpus.

Intuitively, we can say that words with higher TFIDF weight are more important. This intuition is also used for keyword extraction from one or more documents. When extracting keywords from a set of selected documents, we take their sparse vector representations (having TFIDF weights), we sum the vectors, and sort the words according to the TFIDF weight. The keywords are the words with the highest weight in the sorted list. This method is not perfect for selecting the best keywords (again, recall measure is usually higher then precision), but it gives reasonable results, is computationally very efficient and its results are easy interpretable. This method could be understood also as calculating an average document from a set of documents – this average document is also referred to as a centroid vector in the context of clustering (e.g. K-Means algorithm). This method of calculating most representative keywords from a set of documents is related to other eigenvector based methods (such as SVD, PCA, etc.) which are also used to calculate vectors of keywords but are in general computationally much more expensive and in general don’t provide significantly better results. Figure 3 show an example of such a centroid vector for the documents from ACM TechNews corpus which mention the phrase “Semantic Web”.



SEMANTIC (0.548)
SEMANTIC_WEB (0.524)
WEB (0.261)
BERNERS (0.119)
BERNERS_LEE (0.119)
ONTOLOGIES (0.100)
SEARCH (0.099)
LEE (0.099)
W3C (0.094)
RDF (0.089)
WORLD_WIDE_WEB (0.082)
METADATA (0.082)
WORLD_WIDE (0.081)
WIDE_WEB (0.081)
OWL (0.067)
WIDE_WEB_CONSORTIUM (0.065)
WEB_CONSORTIUM (0.065)
LANGUAGES (0.063)

Figure 3. Top 18 keywords with their TFIDF weight for the documents from ACM TechNews that contain phrase "Semantic Web".

5 Visul interface

In this section we present the client part of the "Contexter" system offering graphical user interface to the pre-calculated name-entities and bag-of-words representations of the news documents corpus which are stored together with the original textual representation within the database.

The core part of the system is the main graphical user interface form, which primarily offers two functionalities:

1. Browsing through the network of connected name-entities (two name-entities are connected if they appear in at least one common document).
2. Visualizing a context of a name-entity appearance within the corpus. The context of a name-entity is shown in three different ways:
   - by a set of keywords usually collocated with the selected name-entity,
   - by a set of other name-entities usually collocated with the selected name-entity,
   - by a set of keywords collocated with the simultaneous appearance of the selected and most frequent other name-entities.

Usage of "Contexter" consists from the following steps:

1. Preprocessing of the document corpus which generates name-entity and bag-of-word representations which are saved together with the original textual representation within a database. This step is preformed only once per database change. Since all the algorithms used in the preprocessing phase are computationally efficient, this step takes approx. 15 seconds for the whole ACM TechNews corpus (11.000 articles) on the 2.4GHz PC. We also experimented with other larger corpora (non English languages) and the experiments showed the system scales linearly according to the size of the data (which is expected according to the design of the system).
2. The user runs the client (see Figure 4) with the graphical user interface. First, the user connects to the database that contains the three document representations (see Section 4). This loads a part of the data into the system (list of all name-entities and cashed part of the bag-of-words sparse vectors).
3. As the user selects a name-entity in the left most window (eg., "Marc_Andreessen" in Figure 4), the system instantly shows the corresponding content in other three "context windows". First to the right is the window (1) with the graphical representation of the local context of the network around the selected name entity, (2) next, window to the right shows the context in the form of characteristic keywords from the documents where the selected name entity appears, and (3) the right most window shows the context in the form of the most frequent other name-entities collocated with the selected name-entity.

In the next steps, the user can select other name-entities (either from the complete list on the left, from the graphical interface in the middle or from the right most context list) which instantly adapts the screen according to the new selection. With additional local menu functions the user can view the actual context of the documents where the selected name entities appear.

6 Discussion

In the paper we presented design, architecture and implementation of the system "Contexter" used for analytical browsing of news articles. In the first stage documents from the corpus are preprocessed and transformed into two alternative representations – each document gets in addition to its original textual representation also name-entity and bag-of-words representations. As we are dealing with large amounts of text, for both transformations we decided to use simple and computationally efficient procedures which give satisfactory results in terms of quality. Quality could have been slightly increased with the selection of some other methods, but on the cost of computational efficiency which would further decrease usability of the interface.

There are several potential additions which are interesting for the future development of the system. In particular, with a more detailed analysis of the text in the preprocessing stage using some natural-language-processing tools, we would be able to identify finer grained contexts in which an individual name-entity is appearing; furthermore it would be possible to detect more explicit relationships between the name entities. Next, some more text visualization and text summarization techniques can be applied to extend levels of abstraction when observing the content. With an improved name-entity recognition and consolidation (disambiguation), the usability of the system would increase especially in the cases where the cost of the preprocessing phase (in terms of time and human resources) is not very important.

Finally, the whole system would benefit a lot from a wider Human-Computer-Interaction study which would evaluate current system and suggest corrections to the user interface design and to the needs for various user profiles. In the current stage we designed system mainly for research journalists from some of the Slovenian daily newspapers which contributed suggestions through descriptions of their needs and what they perform in their everyday routine.

Acknowledgement

This work was partially supported by Slovenian Ministry of Education, Science and Sport and SEKT 6FP IP project on semantically enabled knowledge technologies.

References

[1] Geroimenko, V., Chen, C. (ed): *Visualizing the Semantic Web*. Springer Verlag, (2003).

[2] Manning, C., Schütze, H.: *Foundations of Statistical Natural Language Processing*. MIT Press (1999).

[3] Fayyad, U., Grinstein, G., Wierse, A.: *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann (2001).

[4] Chakrabarti, S.: *Mining the Web: Analysis of Hypertext and Semi Structured Data*. Morgan Kaufman (2002).

[5] Jurafsky, D., Martin. J.H.: *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall (2000).

[6] Chen, C.: Visualization of Knowledge Structures, In *Handbook of Software Engineering and Knowledge Engineering*, World Scientific Publishing (2002).

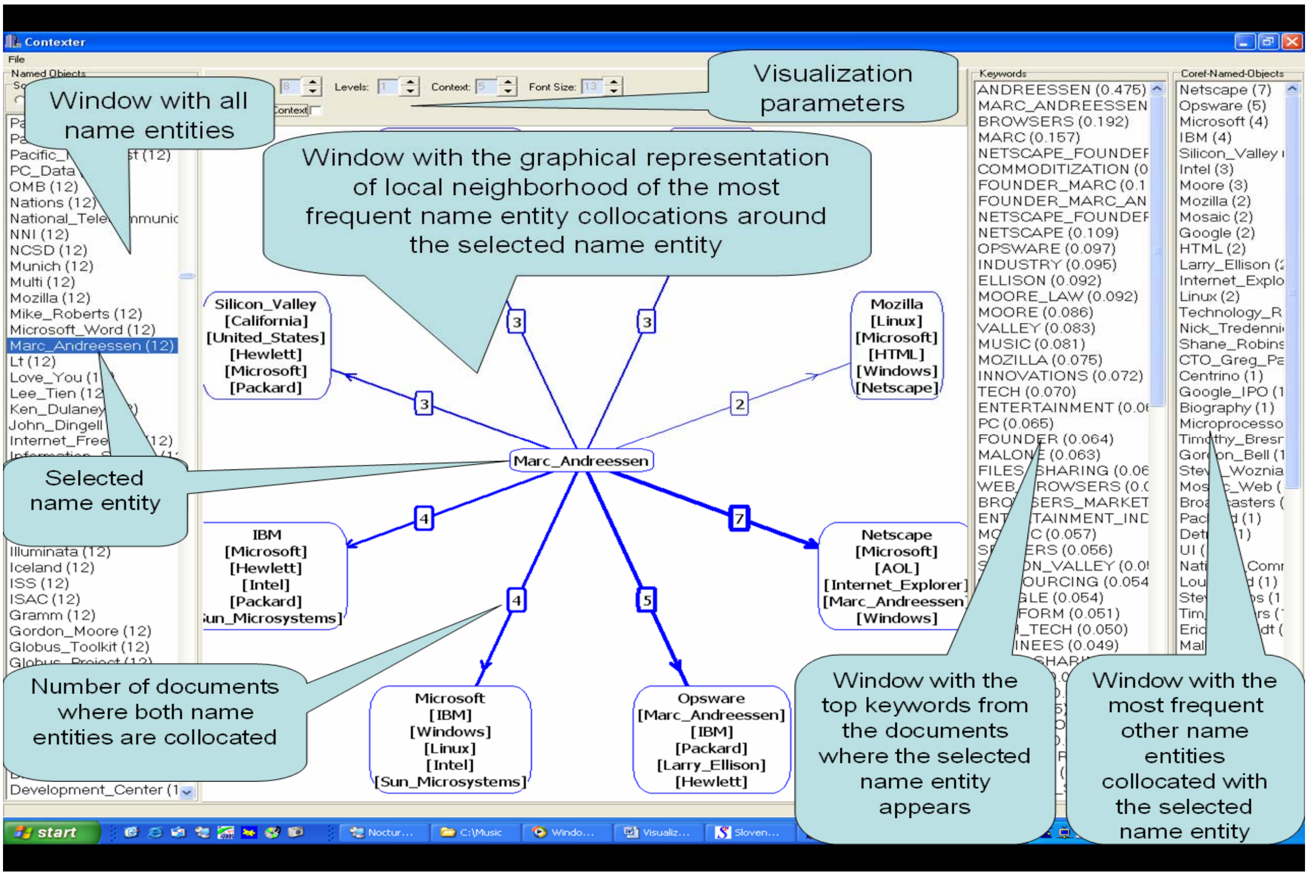
[7] Chase, P., D'Amore, R., Gershon, N., Holland, R., Hyland, R., Mani, I., Maybury, M., Merlino, A., Rayson J.:Semantic Visualization. *ACL-COLING Workshop on Content Visualization and Intermedia Representation*.

[8] Content Visualization and Intermedia Representations (CVIR'98), http://acl.ldc.upenn.edu/W/W98/

[9] Havre, S., Hetzler, E. , Whitney, P., Nowell, L.: ThemeRiver: Visualizing Thematic Changes in Large Document Collections. *IEEE Transactions on Visualization and Comp. Graphics*, V8, No.1, 2002.

[10] Pacific Northwest National Laboratory, Information Visualization, http://www.pnl.gov/infoviz/

[11] Grobelnik, M., Mladenic, D.: Efficient visualization of large text corpora. *7thTELRI, Info. in Corpora* (2002).


Contexter
File
Named Objects
Keywords
Coref-Named-Objects
Levels: 1
Context: 5
Font Size: 13
Context
start
Window with all name entities
Visualization parameters
Window with the graphical representation of local neighborhood of the most frequent name entity collocations around the selected name entity
Selected name entity
Number of documents where both name entities are collocated
Window with the top keywords from the documents where the selected name entity appears
Window with the most frequent other name entities collocated with the selected name entity
Marc_Andreessen
Silicon_Valley [California] [United_States] [Hewlett] [Microsoft] [Packard]
Mozilla [Linux] [Microsoft] [HTML] [Windows] [Netscape]
IBM [Microsoft] [Hewlett] [Intel] [Packard] [Sun_Microsystems]
Netscape [Microsoft] [AOL] [Internet_Explorer] [Marc_Andreessen] [Windows]
Microsoft [IBM] [Windows] [Linux] [Intel] [Sun_Microsystems]
Opsware [Marc_Andreessen] [IBM] [Packard] [Larry_Ellison] [Hewlett]
3
3
3
2
4
7
4
5

Pacific_I st (12)
PC_Data
OMB (12)
Nations (12)
National_Tele mmunic
NNI (12)
NCSD (12)
Munich (12)
Multi (12)
Mozilla (12)
Mike_Roberts (12)
Microsoft_Word (12)
Marc_Andreessen (12)
Lt (12)
Love_You (1
Lee_Tien (12
Ken_Dulaney
John_Dingell
Internet_Free (12)
Illuminata (12)
Iceland (12)
ISS (12)
ISAC (12)
Gramm (12)
Gordon_Moore (12)
Globus_Toolkit (12)
Development_Center (1

ANDREESSEN (0.475)
MARC_ANDREESSEN
BROWSERS (0.192)
MARC (0.157)
NETSCAPE_FOUNDEF
COMMODITIZATION (0
FOUNDER_MARC (0.1
FOUNDER_MARC_AN
NETSCAPE_FOUNDEF
NETSCAPE (0.109)
OPSWARE (0.097)
INDUSTRY (0.095)
ELLISON (0.092)
MOORE_LAW (0.092)
MOORE (0.086)
VALLEY (0.083)
MUSIC (0.081)
MOZILLA (0.075)
INNOVATIONS (0.072)
TECH (0.070)
ENTERTAINMENT (0.0
PC (0.065)
FOUNDER (0.064)
MALONE (0.063)
FILES SHARING (0.06
WEB ROWSERS (0.0
BRO SERS_MARKET
ENT TAINMENT_IND
MC C (0.057)
S ERS (0.056)
ON_VALLEY (0.0
OURCING (0.054
GLE (0.054)
FORM (0.051)
_TECH (0.050)
INEES (0.049)

Netscape (7)
Opsware (5)
Microsoft (4)
IBM (4)
Silicon_Valley
Intel (3)
Moore (3)
Mozilla (2)
Mosaic (2)
Google (2)
HTML (2)
Larry_Ellison (
Internet_Explo
Linux (2)
Technology_R
Nick_Tredenni
Shane_Robins
CTO_Greg_Pa
Centrino (1)
Google_IPO (1
Biography (1)
Microprocesso
Timothy_Bresr
Gordon_Bell (1
Stev _Woznia
Mos _Web (
Broa asters (
Pac d (1)
Det 1)
UI (
Nati Comr
Lou d (1)
Ste s (1
Tim rs (
Eric dt (
Mal

Figure 5. Graphical interface of "Contexter" for browsing/visualizing the name-entity network.