Extracting Named Entities and Relating Them over Time Based on Wikipedia

Abhijit Bhole
Indian Institute of Technology, Bombay, Mumbai - 400076, India
E-mail: abhijit.bhole@cse.iitb.ac.in

Blaž Fortuna, Marko Grobelnik and Dunja Mladenić
Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
E-mail: {blaz.fortuna,marko.grobelnik, dunja.mladenic}@ijs.si

Keywords: text mining, document categorization, information extraction

Received: August 14, 2007

This paper presents an approach to mining information relating people, places, organizations and events extracted from Wikipedia and linking them on a time scale. The approach consists of two phases: (1) identifying relevant pages - categorizing the articles as containing people, places or organizations; (2) generating timeline - linking named entities and extracting events and their time frame. We illustrate the proposed approach on 1.7 million Wikipedia articles.

Povzetek: Predstavljene so metode rudarjenja informacij iz Wikipedie in urejanje v časovno zgradbo.

1 Introduction

Wikipedia is an abundant and valuable source of information manually constructed and mainly targeting human readers, and hence remains unfriendly towards automatic information extraction and mining. The text in Wikipedia is written using a special markup which is mainly aimed towards formatting the text and to a limited extent standardizing pages belonging to same categories. In this paper we propose an approach to extracting information from Wikipedia based on a standard method of first identifying pages that are relevant for the information extraction task and then extracting the desired information as illustrated in Figure 1. We use machine learning methods to identify pages belonging to the same category (in our case person, place or organization) as we have described in [12] and then proceed with text mining of articles to get links and time line information on named entities. The result of our approach is a collection of pages belonging to the predefined categories and a dynamic graph showing named entities (people, places and organizations) and relations between them. For instance, for each person we have important dates from his/her life and some events including places and organizations possibly associated with some dates. Our work is based on standard machine learning and text mining methods [7], in particular for document categorization we use linear support vector machine (SVM) [4], as it is currently considered the state-of-the-art algorithm in text categorization. We used binary linear SVM, the implementation from TextGarden [5]. The main novelty in this work is in extracting dynamic graph relating named entities based on Wikipedia. The closely related work is on semantically annotated snapshot of the English Wikipedia [1]. However that approach is based on natural language processing, while we are using machine learning and mining the extracted data in order to connect the extracted named entities. Another related work known as dbPedia [3] is on extracting structured information from Wikipedia mainly by using the existing structured information, such as Wikipedia *infobox templates* and by making this information available on the Semantic Web .

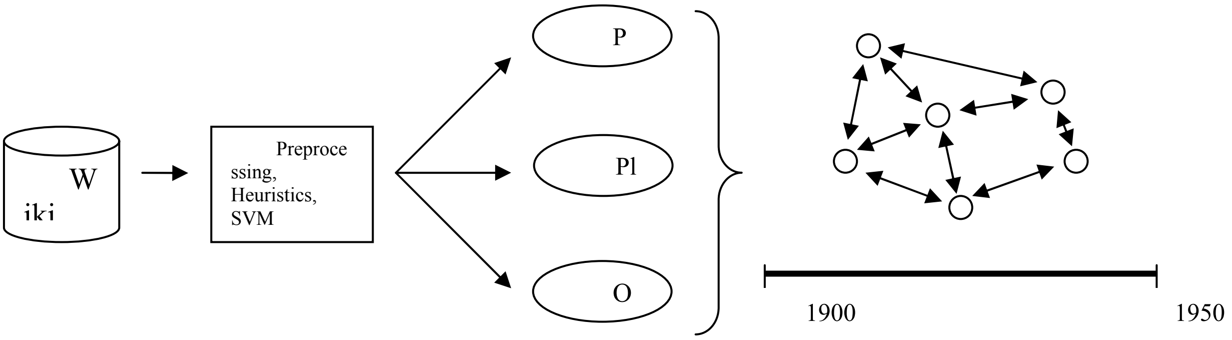


Figure 1. Illustration of the approach consisting of two phases: identifying pages containing named entities and generating timeline.

The task of building timelines from Wikipedia data was first pioneered by the project Wikistory [9], a project which creates a dynamic timeline of scientists based on the data obtained from dbPedia. Somewhat related is also work on Semantic MediaWiki which introduces some additional markup into the wiki-text which allows users to manually add "semantic annotations" to the Wiki [2].

The rest of the paper is structured as follows. Section 2 describes the proposed approach. Section 3 gives experimental results and illustrative example of the obtained dynamic graph. Section 4 concludes with discussion.

2 Approach description

Our goal is to generate a dynamic graph with named entities and relations between them, such as; a person was born at date X at city of Y, in year Z he/she lived in city W. In order to extract named entities we use Wikipedia and first categorize all its pages to find those containing the targeted named entities. We use two approaches to categorizing the pages: one based on manually constructed heuristic rules and the other based on labeling some of the examples and training an SVM classifier.

As Wikipedia pages contain articles with text and some markup information, there are two ways of approaching categorization of documents using heuristics: based on markup and based on text. We can analyze Wiki markup in the articles which can lead to clues for document categorization. However, we should be aware of the fact that the Wiki markup is not semantic annotation and is not uniformly and strictly enforced across the whole Wikipedia. The other approach is to extract plain text from Wiki and then search for particular phrases. We use regular expression based search and replace method throughout for parsing the Wiki markup, obtaining plain text as well as searching for heuristics.

Categorization of Wikipedia pages can be realized either by using index pages on Wikipedia or by classifying each individual articles. Our preliminary experiments using index pages show that it is not a reliable and consistent method over the long term since the index pages are not automatically generated but are maintained by the users themselves. We also tried with filtering some page titles based on tagged word corpuses, but the results were of limited value. Our current strategy involves analyzing individual articles by first using heuristic-based categorization of articles and then, once we have a sufficient number of articles, applying machine learning techniques. A brief description of these heuristics is given along with the experimental results in Section 3.

Once we have categorized Wikipedia pages as describing people, place or organization getting the corresponding named entities is trivial from the Web page URL (e.g., Wikipedia page describing Abraham Lincoln is on URL http://en.wikipedia.org/wiki/Abraham_lincoln). Once we have a collection of articles describing named entities we apply mining for relations between the extracted named entities. We define relation between two entities as follows. An entity is related to other if it has a certain probability of reaching that entity using the Wikipedia hyperlink structure. This usually reflects real life relationships, but inevitably with some difference. The approach we use for finding relations is to build a matrix based on out-links and in-links from a Web page and then searching for two named entities occurring together in the text of other articles.

There are two possible ways of realizing events: one is to identifying articles belonging to events and other is to apply text mining on article texts searching for sentences having mention of dates in them. The task here involves first sentence boundary disambiguation, then extracting dates from these sentences and also linking them to the corresponding named entities.

3 Experiments

3.1 Overview

The entire Wikipedia consists of 5 million entries including redirections, list pages and namespace pages (“Category:”, “Template:” etc), out of which about 1.7 million can be considered as articles describing some unique concept. These 1.7 million articles can be obtained by simple heuristic filtering, which forms our main corpus. Out of that corpus, we selected a random sample of 1000 articles and manually labeled them with three categories (describing people, place, and organization) and left the rest unlabeled. In that sample of articles we found 260 people, 184 places and 118 organizations. We used that for evaluating our approach and estimating the total number of named entities present in the corpus. Although the size of the manually labeled test sample is rather small, it is our hypothesis that it represents the diversity of the source.

Our preliminary experiments identifying named entities through index pages were made by crawling pages with titles similar to “list of people” resulting in about 130K hits which is estimated to result in about 30% recall and about 85% precision. An effort to filter out titles using word corpus failed to have any significant impact on accuracy due to the fact that many persons have fairly common words in their names while some non-person articles may have uncommon words. Similar was true for the other two categories (places, organizations).

3.2 Heuristic based identification

We use parts of Wiki markup known as “Infobox” for identifying some people, places and organizations. We then search for articles with longitude and latitude co-ordinates and annotated them as place while we use the first paragraph from the extracted plain text from the article to search for keywords and dates for identifying people. In this way we labeled 285.000 articles as people, 150.000 as places and 26.000 as organizations. We have evaluated the proposed heuristic approach on our manually labeled sample of 1000 articles. The results are summarized in Table 1.

	Precision	Recall
People	100%	62.4%
Places	95.7%	48.9%
Organizations	100%	10.16%

Table 1: Evaluation of heuristics based classification of Wikipedia articles.

More detailed analysis of the obtained results shows several common sources of error. Many people left out by our heuristic do not have indicative dates in sufficient proximity of the beginning of article. Also the heuristic takes into account most common formats of date, but is conservative enough not to include irrelevant text such as ISBN numbers to maintain its precision. After all, the goal is to apply machine learning with sufficient labeled data rather than to rely entirely on heuristic. When identifying places, several articles on military vehicles or asteroids may have co-ordinates in them but do not qualify for this category which leads to the error. An organization is a loosely defined term, can be any entity which links people (company, university, school or even a rock band). Hence they lack any such indicative text and remain difficult to be found simply by this technique. Some examples of successfully identified named entities are given in Figure 2.

"Aristotle (Greek: Ἀριστοτέλης *Aristotélēs*) **(384 BC – 322 BC)** was a Greek philosopher, a student…"

Example 1: The article describing Aristotle which belongs to category person having birth dates at the beginning of article.

{{Infobox_**Philosopher**
| me = {{polytonic|Ἀριστοτέλης}} "Aristotélēs"
| image_name = Aristoteles Louvre.jpg
| color = #B0C4DE
| region = Western philosophy
| era = [[Ancient philosophy]]
| name = [[Aristotle]]
| birth = [[384 BC]]

Example 2: The article describing Aristotle has a specific Wiki markup indicating him to be a philosopher and thus classifiable as a person.

The coordinates of the nominal centre of London (traditionally considered to be the original [[Eleanor Cross]] at [[Charing Cross]], near the junction of [[Trafalgar Square]] and [[Whitehall]]) are approximately **{{coor dms|51|30|29|N|00|07|29|W|type:city(7,000,000)_region :GB}}.**

Example 3: A co-ordinate header in the article describing London.

Figure 2. Examples of Wikipedia text of some identified named entities for category person (Examples 1 and 2) and place (Example 3).

3.3 Machine learning based identification

In order to apply machine learning, text of the documents to be classified was preprocessed by removing stop-words applying stemming and representing each document using the standard bag-of-words approach containing individual words. Representation of each document was further enriched by frequent phrases, which were considered to be those consisting of up to two consecutive words [6] and occurring at least fifty times in the data collection. The binary classification model was automatically constructed using Support vector machines for each of the three classes, taking the training documents of the class as positive and the training documents of other classes as negative. The classification model was trained on one part of the data collection, leaving the other part to be classified using the standard statistical method called cross validation. In other words, the data collection was randomly divided into several disjoint parts (in our case three) of approximately equal size. Then three classification models were generated, each taking one of the three parts as testing and the remaining two parts as training documents. We report average performance (precision, recall) over the three models. We first ran 3-fold cross validation on manually labeled dataset, using only the first paragraph from the article text (Table 2a) and using the plain text from the entire article (Table 2b).

	Precision	Recall	F1	BEP
People	92.81%	49.20%	64.22%	60.40%
Places	73.14%	54.46%	62.03%	64.88%
Org.*	25.96%	49.17%	33.91%	28.05%

Table 2a: Results from cross validation on sample set using only first paragraphs of article text.

	Precision	Recall	F1	BEP
People	85.18%	63.38%	72.67%	75.57%
Places	85.31%	60.29%	70.46%	77.02%
Org.*	47.63%	37.73%	41.59%	43.11%

Table 2b: Results from cross validation on sample set using plain text from entire article.

*Since the sample set was too unbalanced in this category, the cross validation was run with bias misclassification cost (SVM parameter j=5).

The text extracted from the first paragraph of the articles was used in all subsequent classification experiments presented in this paper since it is significantly computationally less expensive. However, the results obtained using the entire text of the articles (Table 2b) are better than the results of using the first paragraph only (Table 2a). We also tried to use the text including the Wiki markup and it produced results that are a few percent better than those in Table 2b, but this needs further analysis.

The diversity of the sample set can be explained to be a cause of the low recall in case of people and places, where the SVM may have misclassified some article alien to it. However, the precision of people and places was encouraging considered the small size of the sample set as the SVM succeeded in picking up most of the entities which were sufficiently represented in this training data. The poor results with organizations can be explained either by insufficiency of the sample set to capture the diversity of the category or by inability of the bag-of-words text representation to capture all the features necessary for identifying organizations.

We also ran 3-fold cross validation using only the first paragraphs on the whole corpus, which was labeled solely by our heuristics. In the first experiment we use a binary SVM to train the classification model (Table 3a) and in the second experiment we use one class SVM (Table 3b).

	Precision	Recall	F1	BEP
People	83.59%	79.72%	81.61%	82.30%
Places	83.58%	72.19%	77.47%	78.58%
Org.	38.85%	58.28%	46.62%	45.66%

Table 3a: Results of cross validation using binary SVM on the whole corpus

The experiments show that binary SVM is much more successful than one class SVM in partitioning our data. Also, we can assert that our hypothesis that the features present in the text are good enough to enable automatic classification of named entities is correct to a large extent.

	Precision	Recall	F1	BEP
People	46.66%	89.50%	61.34%	63.74%
Places	38.01%	89.39%	53.34%	51.88%
Org.	9.41%	86.32%	16.97%	23.72%

Table 3b: Results of cross validation using one class SVM on the whole corpus.

When running binary SVM on the entire text of articles, we didn't observe considerable improvements over the results given in Table 3a. Namely, we got F1 of 79.37%, 78.79% and 37.09% for people, places and organizations respectively and BEP of 80.14%, 80.05% and 52.01%.

3.4 Analyzing Relations

We analyzed relationships between the total of 285,549 persons for three different kinds of clues relating them: in-links, out-links occurring in the article pages and persons mentioned in the same paragraph of a text of another article or a list. Each of the three clues was given a weight based on preliminary experiments. Then an adjacency matrix ("to-from") of a graph with persons as nodes was constructed and the end result was obtained by sorting the edges by their weight. Further extension is possible by recursively multiplying the matrix in order to consider paths having length more than one [8]. Example result providing relations for Aristotle is given in Figure 3.

In our experiment, about 2.95 in-links / out-links per person were taken into consideration. Apart from that references to pairs of persons occurring together across total 5,735,346 pages were used to assign weights to relations. Figure 4 gives percentage of the number of relations over different weights larger then 0.1. We can see that 46% of relations have weight between 0.1 and 0.2 (578,083 out of 1,258,807).

In addition to associations based on explicit links and co-occurrences, further association is also possible by finding overlaps between time lines of two people. Preliminary analysis on the sample of data has shown that the proposed approach is promising; however a quantitative analysis of these results is part of the future work.

Aristotle is related to:

Plato	(25.217420)
Thomas Aquinas	(4.700384)
Socrates	(4.536786)
Cicero	(3.608422)
Alexander the Great	(3.017379)
Plutarch	(3.011533)
Averroes	(3.000203)
Demosthenes	(2.028392)
Ptolemy	(1.938013)
Aristophanes	(1.848224)
Avicenna	(1.823166)
Galileo Galilei	(1.714287)
Hippocrates	(1.688921)
Euclid	(1.670485)
Homer	(1.659085)

Figure 3: The first 15 persons related to Aristotle are shown along with their corresponding importance weights as suggested by our algorithm.

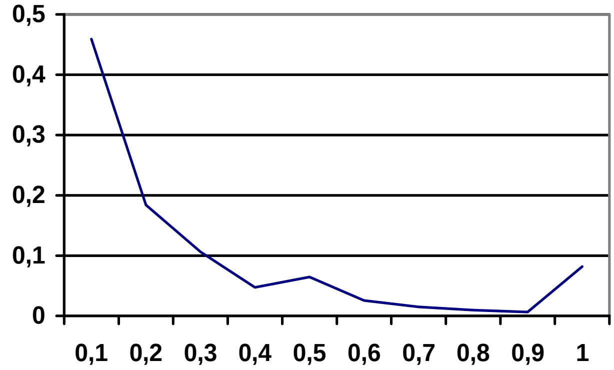
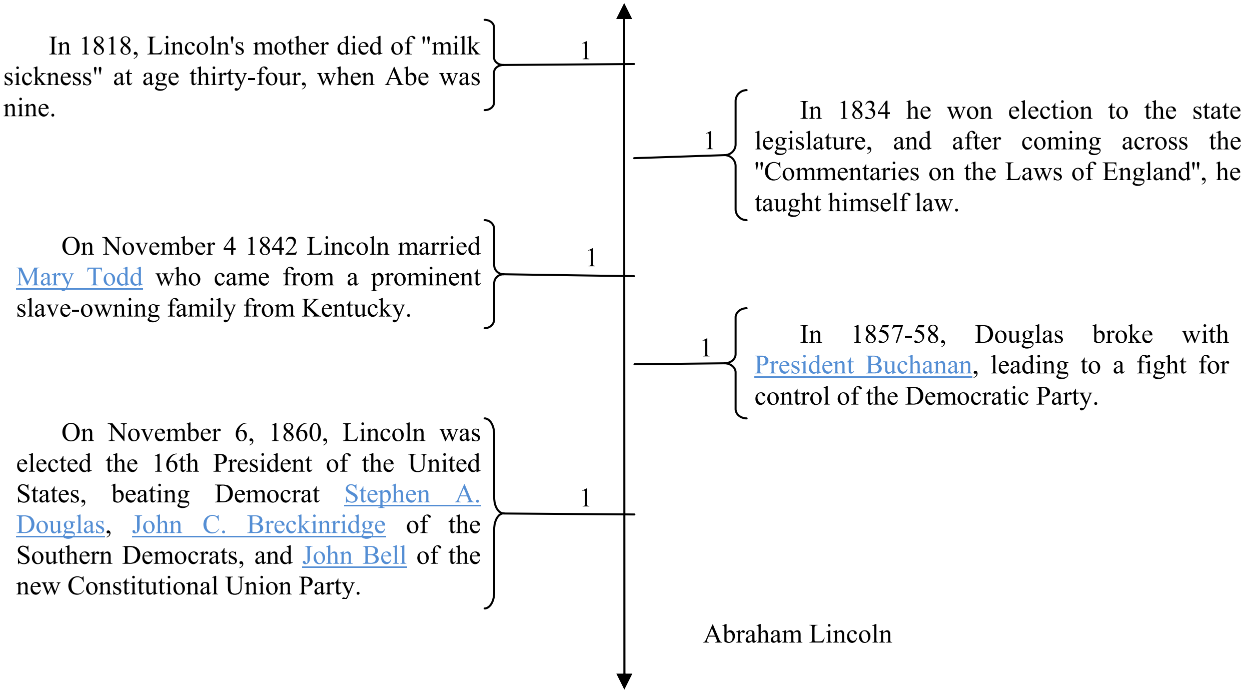


Figure 4: Distribution of relations across different weights.

3.5 Events extraction

To extract events, we used heuristic-based sentence boundary disambiguation after extracting plain text from

Wiki markup and then picked up sentences containing dates. The extracted sentences are regarded as events and are linked to the article in which they were found (see Figure 5 for an example). Other named entities are searched for in the extracted sentences. However the task of incorporating these entities into relationships was already accomplished previously.



In 1818, Lincoln's mother died of "milk sickness" at age thirty-four, when Abe was nine.

In 1834 he won election to the state legislature, and after coming across the "Commentaries on the Laws of England", he taught himself law.

On November 4 1842 Lincoln married Mary Todd who came from a prominent slave-owning family from Kentucky.

In 1857-58, Douglas broke with President Buchanan, leading to a fight for control of the Democratic Party.

On November 6, 1860, Lincoln was elected the 16th President of the United States, beating Democrat Stephen A. Douglas, John C. Breckinridge of the Southern Democrats, and John Bell of the new Constitutional Union Party.

Figure 6: Illustration of timeline generated for Abraham Lincoln showing connections to other identified persons.

Overall we have extracted 2,148,602 events making in average 7.5 per person. In addition we extracted 270,969 birth dates and 113,374 death dates.

- 0/0/-323 - Upon Alexander's death in 323 BC, anti-Macedonian feelings in Athens once again flared.
- 0/0/-86 - When Lucius Cornelius Sulla occupied Athens in 86 BC, he carried off the library of Appellicon to Rome, where they were first published in 60 BC by the grammarian Tyrranion of Amisus and then by philosopher Andronicus of Rhodes.

Figure 5: Example events extracted from the article describing Aristotle.

For the people that we did not manage to obtain birth and death date we used heuristic estimating the time period of the person based on the dates occurring in the events. This heuristics enabled association of named entities based on time lines.

Once we have extracted events and linked them with named entities we generate timeline for a selected named entity that is linking it with other named entities and showing the time frame of their interactions. Illustration of an example generated timelines given in Figure 6.

4 Conclusions

In this paper we outlined how heuristic based approaches can be used for extracting high quality annotations of Wikipedia articles and that automatic text categorization is a viable way of generalizing the heuristics. We have proposed an approach to that consist of two phases: (1) identifying relevant pages containing people, places or organizations and (2) generating timeline linking named entities via the extracting events and their time frame. As a part of future work we are planning to further extend the number of extraction classes, detail level of extracted data from the articles (e.g. type of place, profession of a person, etc.) and investigate integration of the extracted facts into a knowledge base, such as Cyc [10, 11].

Acknowledgement

This work was partially supported by the Slovenian Research Agency and the IST Programme of the EC under NeOn (IST-4-027595-IP) and PASCAL (IST-2002-506778).

References

[1] H. Zaragoza and J. Atserias and M. Ciaramita and G. Attardi. Semantically Annotated Snapshot of the English Wikipedia v.0 (SW0), http://www.yr-bcn.es/semanticWikipedia, 2007

[2] Semantic MediaWiki, http://ontoworld.org/wiki/-Semantic_MediaWiki, 2007

[3] DBpedia.org, http://dbpedia.org/docs, 2007

[4] Joachims, T. Making large-scale svm learning practical. In B. Scholkopf, C. Burges, and A.

Smola, editors, Advances in Kernel Methods, Support Vector Learning, MIT-Press, 1999.

[5] Grobelnik, M., Mladenic, D. 2007. Text Mining Recipes. Berlin, Heidelberg, New York: Springer. Accompanying software available at: http://www.textmining.net.

[6] Mladenic, D., Grobelnik, M. 2003. Feature selection on hierarchy of web documents. Journal of Decision Support Systems, 35. 45-87.

[7] Sebastiani, F. 2002. Machine learning in automated text categorization. ACM Computing Surveys, 34(1). 1–47.

[8] Olston, C., Chi, H. E. (2003). ScentTrails: Integrating browsing and searching on the Web. In ACM Transactions on Computer-Human Interaction (TOCHI), Volume 10, Issue 3, Pages: 177–197, ACM Press, New York, NY, USA.

[9] Wikistory, http://www.urbigene.com/wikistory/

[10] Douglas B. Lenat, (1995). "Cyc: A Large-Scale Investment in Knowledge Infrastructure." Communications of the ACM 38, no. 11, November 1995.

[11] Purvesh Shah, D. Schneider, C. Matuszek, R.C. Kahlert, B. Aldag, D. Baxter, J. Cabral, M. Witbrock, J. Curtis, (2006). Automated Population of Cyc: Extracting Information about Named-entities from the Web. In Proceedings of the Nineteenth International FLAIRS Conference, pp. 153-158, Melbourne Beach, FL, May 2006.

[12] Bhole, A., Fortuna, B., Grobelnik, M., Mladenić, D. Mining wikipedia and relating named entities over time. In: Bohanec, M., Gams, M., Rajkovič, V., Urbančič, T., Bernik, M., Mladenić, D., Grobelnik, M., Heričko, M., Kordeš, U., Markič, O. Proceedings of the 10th International Multiconference on Information Societz IS 2007, 8.-12. october 2007, volume A. Ljubljana: "Jožef Stefan" Institute, 2007, pp. 177-180.