Mining actions from reports on flood

Luboš Popelínský, Jan Blaťák

Knowledge Discovery Lab
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00, Brno, Czech Republic
{popel,xblatak}@fi.muni.cz

Abstract

This paper focuses on mining in short reports that describe a situation in a given area and actions performed as reaction to that situation. Such texts are frequent in crisis management in situations like earthquake, fire or flood. For further analysis it is necessary to filter the relevant pieces of text. We found that common machine learning algorithms fail for filtering such sentences. We describe a novel method based on inductive logic programming which yields in high precision and recall. This method has been successfully used for analysis of reports on flood in Central Europe in 2002. We also discuss different domain knowledge and also various natural language processing tools that we used for preprocessing the documents.

Učinki rudarjenja po poročilih o poplavah

Članek se osredotoča na rudarjenje po dokumentih, ki opisujejo razmere v določenem območju in delovanje kot posledico tovrstnih razmer. Taka besedila so pogosta v kriznem menedžmentu, v razmerah, kot so potresi, požari ali poplave. Za nadaljno analizo je potrebno filtrirati določeno informacijo. Pri razvrščanju besedil se ponavadi dobro obnesejo algoritmi strojnega učenja, kot je naivni Bayesov klasifikator. Ugotovili smo, da pri filtriranju stavkov, ki opisujejo delovanje, ti algoritmi niso uspešni. Opišemo novo metodo, ki temelji na induktivnem logičnem programiranju in daje rezultate z visoko točnostjo in pokritjem. Metoda je bila uspešno uporabljena pri analizi poročil o poplavah v Srednji Evropi l. 2002. Prav tako razpravljamo o različnih specializiranih znanjih in orodjih za obdelavo naravnega jezika, ki smo jih uporabili pri procesiranju dokumentov.

Keywords text filtering, information extraction, term extraction

1. Text mining in crisis management

Exploratory data analysis in geographical domains should not be limited only to data with explicit spatial and temporal information. As bigger and bigger data sources contain data different from that in geographic information systems – e.g. text, hypertext, audio and video sequences, it is necessary to look for tools that have been developed for this kind of data and adapt them for specific purposes in geographical domains.

In crisis management, like flood, earthquake or fire management, a big amount of messages and reports is being exchanged between the parties that participate in the recovery process. Any tool that decreases this amount or even extract the relevant information can be helpful. An example is text filtering (Blaťák and Popelínský, 2004a; Sebastiani, 2002) where each document is classified into one of two classes, e.g. `INTERESTING` and `NON-INTERESTING`. When using such a tool, the recipient obtains only the relevant messages or messages relevant with a high confidence. In (Popelínský and Blaťák, 2006) we showed that methods based on the state-of-the-art propositional learning techniques can reach high accuracy when classifying whole document or a document paragraph.

However, in all these experiments whole document was supposed to belong to one class. Unfortunately it is not the case in reality because messages may contain short pieces of text with information on different topics. E.g. in the case of reports on flood a message consists of description of a current situation as well as description of actions performed. In (Popelínský and Blaťák, 2006) it was demonstrated that good performance can hardly be reached with propositional learning algorithms like Naive Bayes or Support Vector Machines without user intervention, namely without new features construction. One reason is the poor language for building the classifier which is actually built upon propositional logic only. Another reason is the small length of the information that are to be filtered – one sentence, one clause in a sentence or even a subpart of a clause.

In this paper we show that knowledge-intensive learning techniques, namely inductive logic programming (Cussens and Džeroski, 2000; Džeroski and Lavrač, 2001) that exploits predicate logic, can help to solve this problem. We aim at building a tool that gives a trustful answer to some of classification queries and maybe leaves some queries unanswered. The main goals of this work were

- to find an appropriate representation for this kind of tasks
- to find feasible natural language processing tools for pre-processing the text data and for enriching domain knowledge
- and eventually to find a method that reach high precision

Domain knowledge contain, for each word, information about its position in the sentence, a part-of-speech tag, a syntactic category and also hyperonymas in a domain-dependent ontology.

We demonstrate our approach on processing reports on flood in Central Europe in 2002. The problem is displayed in Section 2. The data used in experiments are introduced in Section 3. In Section 4. we introduce natural language processing (NLP) tools that we used for text pre-processing. Section 5. contains description of data transformations and several variants of domain knowledge. Description of the method can be found in Section 6. and results in Section 7. We conclude with discussion in Section 8. and with plans for future work in Section 9.

2. Reports on flood

News reports on flood, like the example below

> *In the Czech Republic the capital Prague is bracing for a major flood, just days after storms in the south of the country killed six people. "The forecast is bad," said Josef Novotny of the Prague crisis committee, warning that the Vltava river could burst its banks overnight. Floods affected some parts of Prague on Friday, but Mr Novotny said twice as much water was now bearing down on the city. Several southern towns are already cut off by water, and some have been evacuated. "Trains are not running, because bridges have fallen, and buses are not running, because roads are damaged," the mayor of the southern town of Prachatice, Jan Bauer, told Czech radio. Officials called on residents of the UNESCO-protected town of Cesky Krumlov – the second most popular tourist destination in the country – to leave.*
>
> *(Radio BBC Archive)*

usually contain two kinds of information. The first one concerns description of the current situation, the other describes an action performed, e.g. by an emergency unit. For instance the sentence

> *In the Czech Republic the capital Prague is bracing for a major flood, just days after storms in the south of the country killed six people.*

describes a situation whilst the sentence

> *Officials called on residents of the UNESCO-protected town of Cesky Krumlov – the second most popular tourist destination in the country – to leave.*

an action. It is evident that a sentence (or more generally, a part of the message) can concern both, or be irrelevant. Then the goal of a classification can be defined as an assigning a label from the set {`SITUATION, ACTION, BOTH, IRRELEVANT`} to each part of the given news report. The class `BOTH` contains sentences that concern both the current situation and the action performed. Then the label `IRRELEVANT` is assigned to all sentences that cannot be classified to none of these classes because the sentence brings no information relevant to a situation or to an action.

This work is the first step to fully understand such kind of reports. If we know that a sentence concerns, e.g., an action, a goal of the next step is understanding this action, e.g., learning the subject – agent(s) and target(s) or spatial and temporal relations. Such knowledge can be then used directly for decision support.

3. Data

In our experiments we used the summary report on flood in 2002 that has been manually collected (Andrienko, 2001). For each day there are two paragraphs, one describing the situation in the region affected with flood and the other referring about actions performed. The part of the description of the first day of the flood follows.

> *9 August 2002*
> *Situation*
> *Unusually heavy rains falling over a broad area of Central Europe have resulted in widespread flooding. In Austria, Bulgaria, the Czech Republic and Romania the floods have been particularly severe. The weather forecast for the next few days threatens even more rain. A rain dense and very slow moving front is lingering over the area, heading toward the Black Sea*
> *...*
> *Actions*
> *In Austria, the Red Cross has been working together with the fire brigade and the military to aid those affected by the floods. A 24 hour around the clock operation helped to ensure that those at risk were rescued. While efforts are continuing, it is believed that all of those who were in immediate danger have now been assisted. However, water levels remain dangerously high, with the risk of more rain at any moment. The Red Cross also organized mobile kitchens, providing hot food and drinks to those affected.*

This report was collected from texts on web – BBC, CNN, France Press, Reuters, Deutsche Welle, The Associated Pres Situation reports of OCHA (United Nations Office for the Coordination of Humanitarian Affairs), ReliefWeb, Emergency appeals and reports of humanitarian organizations: Salvation Army, Red Cross, a report of ENVIS – the Prague Information System on the Environment and an event report of RMS – Risk Management Solutions, Inc.

4. NLP tools

Memory-based shallow parser Memory-based shallow parser (MBSP) (Daelemans and van den Bosch, 2005) splits each sentence into chunks – name phrases, verb phrases or prepositional phrases. Moreover it can recognize borders of the subject and the object part in the sentence. Memory-based part-of-speech tagger that is a part of MBSP returns for each word its morphological category.

Topic maps We also used topic maps, namely Ontopoly from Ontopia[1] for building ontology for actions in flood management. We grouped all terms (mostly one- or n-words noun phrases) into classes of terms and also defined associations between these terms and verbs that appeared in the documents. In the work reported here we exploited only the hierarchy of terms. For each term, we add a pointer to its hyponym or to ANY. The list of classes that contain more than one term consists of *accessories, actions, area, authorities, chemical, doing, impulse, mobileEquipment, organization, state, valuables.*

WordNet Besides the hand-coded hierarchy mentioned above we also employed the WordNet semantic lexicon[2], namely synsets and collection of hyponyms. We generated for each word in documents (not only for the terms) its synset code(s) and its hyponyms.

5. Data representation and domain knowledge

5.1. Data representation

Each document has been morphologically and syntactically tagged with the memory-based shallow parser and then transformed into three relations

```
word(SiD, WordOrder, Word)
tag(SiD, WordOrder, PartOfSpeechTag)
chunk(SiD, WordOrder, Chunk)
```

where `SiD` is the unique sentence identifier and `WordOrder` identifies the position of a word in the sentence `SiD`. This *flat data representation* is then used in the domain knowledge predicates described below.

5.2. Domain knowledge

We use the term "domain knowledge" in the way commonly used in machine learning or inductive logic programming as knowledge that is not or cannot be expressed by learning examples themselves. This notion is more general than a feature description language which actually transforms data into propositional form. In domain knowledge predicates we are capable to describe any dependency between variables in those predicates without explicit building a feature for each dependency.

In (Blaťák, 2005) we described two different sets of background knowledge predicates for text documents, $\mathcal{B}^1$ and $\mathcal{B}^2$. They consist of predicates which specify general properties of a given focus word (*focusWord/2*), for example, that a given position in the sentence is a punctuation (*isPunct/2*), a quotation mark (*isQuot/2*) or that the first letter is capital (*begCap/2*).The difference between $\mathcal{B}^1$ and $\mathcal{B}^2$ lies in a manner of exploring the context of the analyzed word. $\mathcal{B}^1$ uses a literal *hasWord/3* whose first argument determines the relative position of a word with respect to the focus word (e.g. -3 means the third word to the left). The background knowledge $\mathcal{B}^2$ does not use information about a position of a word in the sentence and only introduces an arbitrary word from a context.

For a need in this work we extended $\mathcal{B}^2$ with temporal logic. Each sentence is seen as a sequence of events – words. $\mathcal{B}^3$ domain knowledge thus consists of all predicates in $\mathcal{B}^2$ and temporal predicates

```
follows(SiD,W1,W2)
after(SiD,W1,W2)
precedes(SiD,W1,W2)
before(SiD,W1,W2)
```

that have the meaning "in the sentence `SiD`, word `W2` immediately follows/is after/immediately precedes/is before the word `W1`". An example of a formula in $\mathcal{B}^3$ is below.

focusWord(S,B), after(S,B,C), begCap(S,C), hasTag(S,C,'NNP'), after(S,C,D), hasTag(S,D,'CC').

in the sentence A, there is a word B,
somewhere on the right there is the word C which
starts with a capital letter
and has tag 'NNP'
and somewhere right from the word C there
is the word D with tag 'CC'

Example:
"...[between/IN]^B the/DT United/NNP ^{C} States/NNP and/CC^D China/NNP ..."

6. Experiments

6.1. Aleph

The Aleph[3] is an ILP learner that can learn from noisy data. It chooses one or more positive examples from a training set and constructs their least general generalizations – so called a bottom clause – with respect domain knowledge. Then using literals in the bottom clause, Aleph builds new rules in general-to-specific manner and employs a covering paradigm: it learns one clause a time and after finding it, Aleph removes all positive examples covered by this clause. This repeats until all (but a small fraction of) positive examples are covered and none (but a small fraction of negative) examples are not covered. The degree of incorrectness and inconsistence is driven by user-defined threshold. parameters.

6.2. Description of the method

As positive examples we used sentences that describe an action, the rest has been used as negative examples. Each sentence was enriched with output from memory-based morphological tagger and shallow parser. Further we added the information form hand-coded ontology and information from WordNet - synsets and hyponyms for each word.

The goal was to find a definition of the predicate `s(SiD, Subj, Verb, Obj)`. Arguments of the predicate `s(SiD, Subj, Verb, Obj)` brings information about the sentence (SiD, sentence identifier), a noun that appears in the subject part (Subj), a non-auxiliary verb (Verb), and a noun that appears in the object part (Obj). In

[1] `http://www.ontopia.net/`
[2] http://wordnet.princeton.edu/
[3] http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/

general, there can be more then one learning example per sentence: it may happen e.g. when the subject part contains more than one noun.

The average number of literals was 127.29 (standard deviation 44.95, max 222, min 57).

We used Aleph for finding all rules that cover a minimal number (between 5 and 25) of positive examples in the learning set[4] and then used these rules for classifying unseen test data. The bottom limit was set to 5 because coverage smaller than 5 examples resulted in over-fitting. We used 200, 300, 400, 500 and 600 examples for learning, the rest for testing. The clause length (number of literals in the rule) varied from 3 to 6.

For description of results we use the usual characteristics, precision, recall and the F-1 measure.

All experiments were performed on AMD Athlon™ XP 2500+ computers with 756 MB of memory.

7. Results

Summary of results Precision and recall for different cardinality of learning set are displayed in Fig. 1 and Fig. 2. On X-axis, minpos stand for the minimal coverage. On Y-axis there is precision and recall, respectively. All other characteristics for the case of 500 learning examples are in Table 1.

The fact that precision is increasing with increasing number of learning examples (see Figures 1 and 2) is not surprising. More important is the fact that for 400, 500 and 600 examples differences in precision are very small.

The most important result is the fact that even more significant increase of precision has been observed for increasing minimal coverage. Minimal coverage is the minimal number of positive examples from the learning set that has to be covered by each rule.. When looking at Table 1 it is true that precision for more than 300 examples in the learning set, is always high. But there are also many situations that are incorrectly classified – recall for situations is high. From this respect, the best choice will be higher minimal coverage of rules. We can see that for minimal coverage=22 the recall for situations is half of that for lower values of mininimal coverage.

min_cov.		Prec. (%)	Rec. (%)	F-1 (%)	Acc. (%)
5	act.	87.69	49.31	63.12	56.12
	sit.	32.48	22.11	26.31	
10	act.	88.05	52.69	65.93	58.52
	sit.	33.80	22.85	27.27	
18	act.	89.28	55.08	68.13	60.75
	sit.	35.47	21.13	26.48	
22	act.	93.56	51.38	66.36	60.28
	sit.	36.35	11.30	17.24	

Table 1: Learning from 500 examples

[4]This is called a minimal support in learning association rules

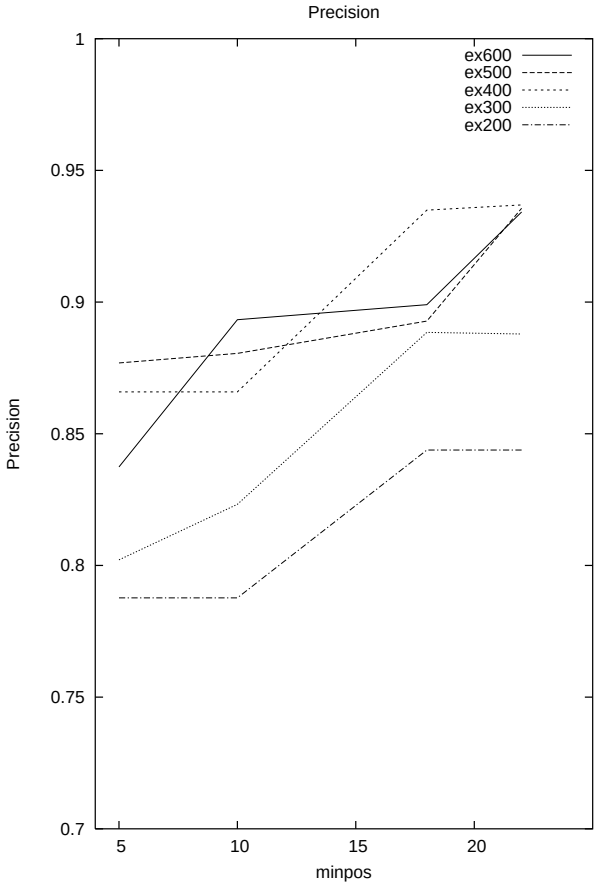

Figure 1: Precision for different learning sets

Rules Examples of the most interesting rules (600 examples in the learning set, clause length=5) are in Fig. 3.

8. Discussion

Best parameters settings We observed that the best clause length was 5 literals. Longer rules did not result in a significant increase of precision.

Dependency on the domain knowledge We also checked how the precision is influenced by the domain knowledge – $\mathcal{B}^1, \mathcal{B}^2$, and $\mathcal{B}^3$ – used. Not surprisingly, precision is increasing with the complexity of the domain knowledge. The same trend, but much more faster, has been observed for recall.

Use of WordNet The use of data from WordNet did not result in increase of accuracy. Information about a synset did not appear in the learned rule at all. Info on hyperonyma appears in less then 5% of rules, and always together with hyperonyma from the hand-coded ontology. It is obvious because the hand-coded ontology is domain-specific and contains more information specific to our task.

State-of-the-art Up to our knowledge, this is the first work on classification of short texts and action recognition. Technically, it is of course a part of the research stream on text filtering (Sebastiani, 2002). Similar goals are solved in the series of workshops on Event Extraction and Synthesis. See e.g. `http:\\www.ics.uci.edu/~ashish/ee.htm`.

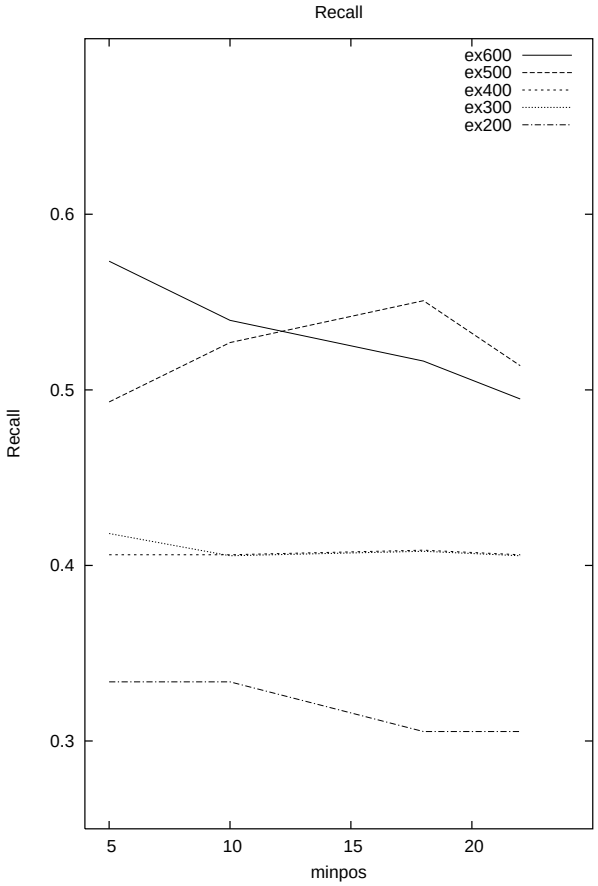

Figure 2: Recall for different learning sets

9. Conclusion and future work

We developed and experimentally confirmed a novel method for filtering small pieces of text that is based on inductive logic programming framework. In filtering sentences that brings information abour actions during floods, the precision overcome 90%.

In future, we want to use this method for term recognition. First results, with propositional learning algorithms has been introduced in (Popelínský and Blaťák, 2006). First-order logic rules learned with Aleph contains even more information. Another way is to use frequent patterns (Blaťák and Popelínský, 2004b) (also called large itemsets) for finding new features. We also plan to exploit other relations defined in the Topic Maps ontology.

We believe that this work can be helpful in automatic information extraction in the process of crisis management. As a small step to understanding the contents of a message, our approach can help to find an equilibrium between a need of understanding and necessary formalization of messages.

Acknowledgement

We thanks Natalia Andrienko for providing the report on flood and Petr Výmola for building the ontology. This work has been partially supported by the Faculty of Informatics, Masaryk University in Brno and by the Grant Agency of the Czech Republic under the Grant No. MSM0021622418 Dynamic Geo-visualization in Crisis Management.

10. References

N. Andrienko. 2001. A report on flood in central europe 2001. Manuscript.

J. Blaťák and L. Popelínský. 2004a. Fragments and text categorization. In Blanche P. and Rodrigues H., editors, *Proceedings of the ACL-2004 Interactive Posters/Demonstrations Session, Barcelona 2004.*

J. Blaťák and L. Popelínský. 2004b. Mining first-order maximal frequent patterns. *Neural Network World*, 5:381–390.

Jan Blaťák. 2005. First-order frequent patterns in text mining. In *Proceedings of the 12th Portuguese Conference on Artificial Intelligence, EPIA'05*, pages 344–350. Institute of Electrical and Electronics Engineers, Inc., December.

J. Cussens and S. Džeroski. 2000. *Learning Language in Logic*. Springer-Verlag.

W. Daelemans and A. van den Bosch. 2005. *Memory-based language processing*. Cambridge University Press.

S. Džeroski and N. Lavrač. 2001. *Relational Data Mining*. Springer-Verlag, September.

L. Popelínský and J. Blaťák. 2006. Mining situations and actions from news. In *Proceedings of Znalosti'06, Czech-Slovak Conference on Artificial Intelligence*, pages 1–3, Feb.

F. Sebastiani. 2002. Machine learning in automated text categorization. In *ACM Computing Surveys*, volume 34, pages 1–47, March.

Rule 1 Pos cover = 128 Neg cover = 0
s(A,B,C,D) :- hasWord1(also,A,E), hasWord1(to,A,F).

Rule 2 Pos cover = 83 Neg cover = 0
s(A,B,C,D) :- precedes(A,D,E), before(A,E,F), isPoS(A,F,'VB'), isVP(A,E).

Rule 3 Pos cover = 184 Neg cover = 0
s(A,B,C,D) :- hasWord1(actions,A,E), before(A,E,F), isPoS(A,F,'NNS'), isPoS(A,E,'NN').

Rule 4 Pos cover = 40 Neg cover = 0
s(A,B,C,D) :- before(A,D,E), isPoS(A,E,'VBZ'), before(A,C,F), isPoS(A,F,'RP').

Rule 5 Pos cover = 24 Neg cover = 0
s(A,B,C,D) :- precedes(A,B,E), isString(A,E,of), before(A,E,F), isPoS(A,F,'VBG').

Rule 6 Pos cover = 174 Neg cover = 0
s(A,B,C,D) :- hasWord1(leave,A,E).

Rule 7 Pos cover = 124 Neg cover = 0
s(A,B,C,D) :- hasWord1(have,A,E), hasWord1(city,A,F).

Rule 8 Pos cover = 49 Neg cover = 0
s(A,B,C,D) :- begCap(A,B), precedes(A,B,E), isString(A,E,were).

Rule 9 Pos cover = 152 Neg cover = 0
s(A,B,C,D) :- hasWord1(12,',',A,E), before(A,C,F), isPoS(A,F,'JJ'), isOBJ(A,F).

Rule 10 Pos cover = 128 Neg cover = 0
s(A,B,C,D) :- hasWord1(popular,A,E).

Rule 11 Pos cover = 59 Neg cover = 0
s(A,B,C,D) :- before(A,B,E), isSBJ(A,E), precedes(A,D,F), isPoS(A,F,'NNS').

Rule 12 Pos cover = 126 Neg cover = 0
s(A,B,C,D) :- hasWord1(12,',',A,E), hasWord1(city,A,F), isPoS(A,B,'NNS').

Rule 13 Pos cover = 83 Neg cover = 0
s(A,B,C,D) :- hasWord1('Prime',A,E).

Rule 14 Pos cover = 125 Neg cover = 0
s(A,B,C,D) :- hasWord1(medieval,A,E).

Figure 3: Rules