Fuzzy Classification of Web Reports with Linguistic Text Mining

Jan Dědek		Peter Vojtáš

***Abstract*— In this paper we present a fuzzy system which increases automation of fuzzy classification of textual web reports. Our approach is based on usage of third party linguistic analyzer and fuzzy inductive logic programming. Main contributions are formal models, prototype implementation and evaluation of experiments of the whole system.**

I. Introduction

General intro

Priklad

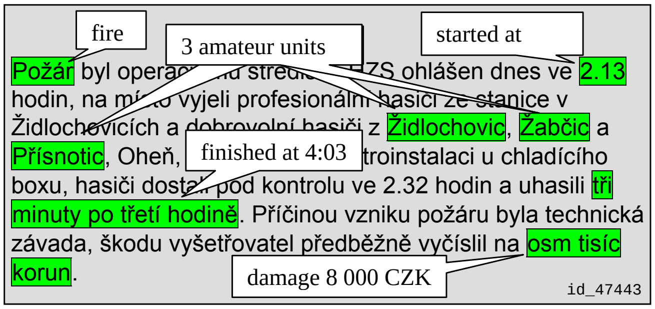

Požár byl operačnímu středisku HZS ohlášen dnes ve 2.13 hodin, na místo vyjeli profesionální hasiči ze stanice v Židlochovicích a dobrovolní hasiči z Židlochovic, Žabčic a Přísnotic, Oheň, [illegible] troinstalaci u chladícího boxu, hasiči dostali pod kontrolu ve 2.32 hodin a uhasili tři minuty po třetí hodině. Příčinou vzniku požáru byla technická závada, škodu vyšetřovatel předběžně vyčíslil na osm tisíc korun.

Fig. 1. Example of analyzed web report.

Problem

Contribution

Related work

more on text mining, information extraction, fuzzy internet applications, ...

In [**?**]

Paper is organized as follows ...

II. Models, methods, design of the system

A. Linguistic Analyser

Citace tectogrammatical structure [1]

B. Classical ILP

In our application we are facing the challenge of induction and/or mining on several places. First we need an inductive procedure when extracting from web texts attributes of an accident.

Second we need an inductive procedure when trying to explain degree of seriousness of an accident by attributes of this accident (also called background knowledge).

Both places where induction has to be used have following requirements

- data are/can be fuzzy
- background knowledge is multirelational
- classification is fuzzy

Having in mind these requirements we chose Fuzzy inductive logic programming. To make the paper readable we present bellow short description of ILP techniques.

In our presentation of Inductive Logic Programming (ILP)

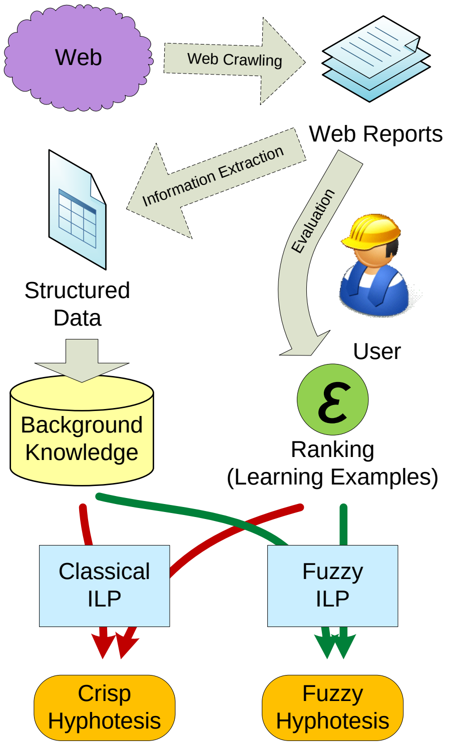
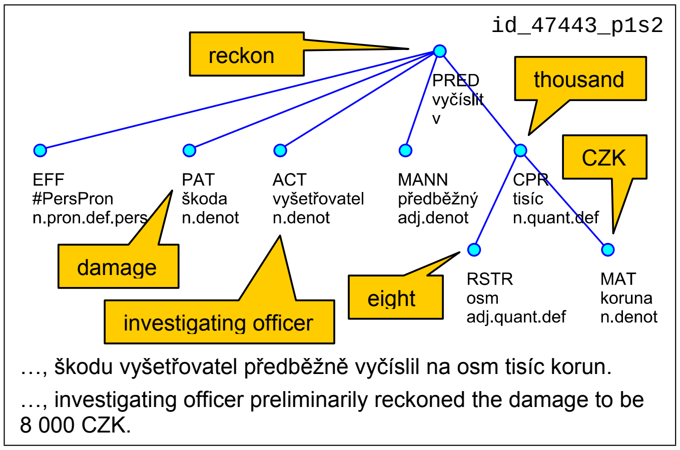


Fig. 2. Schema of presented work (text part).



…, škodu vyšetřovatel předběžně vyčíslil na osm tisíc korun.

…, investigating officer preliminarily reckoned the damage to be 8 000 CZK.

Fig. 3. Example of linguistic tree of one of analyzed sentences.

Jan Dědek is with the Department of Software Engineering, Charles University in Prague, Czech Republic (email: jan.dedek@mff.cuni.cz).

Peter Vojtáš is with the Institute of Computer Science, Czech Academy of Sciences and Department of Software Engineering, Charles University in Prague, Czech Republic (email: peter.vojtas@mff.cuni.cz).

we follow the book of S. Džeroski and N. Lavrač [2]. Citace ILP [3]

Given is a set of examples $E = P \cup N$, where P contains positive and N negative examples, and a background knowledge B. The task is to find a hypothesis H such that

$$(\forall e \in P)(B \cup H \models e)$$

and

$$(\forall e \in N)(B \cup H \not\models e).$$

Typically, E consists of ground instances of the target predicate which has to be classified - in our case accidents. B typically consists of several predicates (relational tables) which describe properties of object which have to be classified - in our case properties of accidents. Background knowledge can contain also some rules. Hypothesis H typically consists of logic programming rules, which when added to B, explain all positive examples and no negative examples.

Main advantage of ILP is it's multirelational character, namely B can reside in several tables.

C. Fuzzy and GAP induction

In our presentation of Inductive Logic Programming (ILP) we follow the Paper of T. Horvath and P. Vojtas [4] about fuzzy inductive logic programming. Citace Fuzzy ILP [4]

We use the approach of the fuzzy logic in narrow sense developped by J. Pavelka and P. Hajek. Formulas are of the form φ, x (φ is syntactically same as in the crisp case) are graded by a truth value $xin[0,1]$. A structure $\mathcal{M}$ consist of domain M and relations are interpreted fuzzy (we do not consider function symbols here). Evaluation $\|\varphi\|_{\mathcal{M}}$ of a formula φ uses truth functions of many valued connectives (our logic is extensional and/or truth functional). Satisfaction is defined by

$$\mathcal{M} \models_f (\varphi, x) \ iff \ \|\varphi\|_{\mathcal{M}} \geq x$$

Given is a fuzzy set of examples $\mathcal{E} : E \longrightarrow [0,1]$ and a fuzzy background knowledge $\mathcal{B} : B \longrightarrow [0,1]$. The task is to find a fuzzy hypothesis $\mathcal{H} : H \longrightarrow [0,1]$ such that

$$(\forall e, f \in dom(E))(\forall \mathcal{M})(\mathcal{M} \models_f B \cup H)$$

we have

$$E(e) > E(f) \Rightarrow \|e\|_{\mathcal{M}} \geq \|f\|_{\mathcal{M}}.$$

That is, it cannot happen that

$$E(e) > E(f) \wedge \|e\|_{\mathcal{M}} < \|f\|_{\mathcal{M}},$$

or rephrased, if E is rating e higher than f, then it can not happen in a model of $B \cup H$ that e is rated worse than f.

Typically, $\mathcal{E}$ consists of ground instances of the target predicate which are classified in truth degrees - in our case degree of seriousness of an accident. B typically consists of several fuzzy predicates (fuzzy relational tables) which describe properties of object which have to be classified - in our case fuzzy properties of accidents - degree od injury, degree of damage, ... Background knowledge can contain also some rules, so far only crisp rules are used. Hypothesis H typically consists of a fuzzy logic program, which when added to B, prevents of misclassification (better can not be declared to be worse, nevertheless can be declared as having same degree (for more detailed discussion on this definition of fuzzy ILP we reffer to the paper [4])). Moreover, in practise, we use GAP - Generalized Annotated Programs, so graded formulas wil lbe sometimes understood as annotated (with crisp connectives and more complex annotation of head of rules.)

D. Translation of fuzzyILP task to several classical ILP tasks

As far as there is no implementation of fuzzy (GAP) ILP, we have to use a classical ILP system. Fortunately a fuzzy ILP task can be translated to several crisp ILP tasks (subject to some rounding and using finite set of truth values).

Assume, our fuzzy sets take values for a finite set of truth values $\{0,1\} \subseteq T \subseteq [0,1]$. For each predicate $p(x)$ in B we add an additional attribute for truth value $p(x,t)$. We construct a crisp background knowledge $\mathcal{B}_T^{mon}$ by a process called monotonization, as follows:

If $\mathcal{B}(p(x)) = t \in T$, then for all $t' \in T, t' \leq t$ we add $p(x,t') \in \mathcal{B}_T^{mon}$.

For all $t \in T$ we create a crisp example set $E_t = P_t \cup N_t$, where

$$e \in P_t \ \ iff \ \ \mathcal{E}(e) \geq t$$

and N_t is the rest.

For each $t \in T$ we create a crisp ILP task $E_t, \mathcal{B}_T^{mon}$ and get a hypothesis H_t guaranteeing examples of degree at least t. Note that variable boundings ib B have no boundings on truth value attribute, which was added to each predicate, and hence there are no variable boundings in H on truth value attribute. To predicate in E we did not add the additional truth value attribute

Let us assume C is the target predicate in the domain of $\mathcal{E}$. We define $\mathcal{H}$ with domain consisting of one GAP rule

$$C(y) : u(x_1, \ldots, x_m) \leftarrow B_1 : x_1 \& \ldots \& B_n : x_m,$$

here $B_1 : x_1 \& \ldots \& B_n : x_m$ is enumaration of all predicates in B.

Assume $B_1(y_1, t_1), \ldots, B_n(y_n, t_n)$ are some predicates in B (for simplicity enumerated from 1 to $n \leq m$). Then for each rule

$$R = C(y) \Leftarrow B_1(y_1, t_1), \ldots, B_n(y_n, t_n)$$

in H_t we give a constraint in definition of u as follows

$$U_R = u(x_1, \ldots, x_m) \geq t \text{ if } x_1 \geq t_1, \ldots, x_n \geq t_n.$$

Note that $x_{n+1}, \ldots, x_m$ have no restrictions.

We claim, that if all H_t were correctly learned by an crisp ILP system then for u the minimal solution of all constraints U_R for all $R \in H_t$, for all $t \in T$, the rule

$$C(y) : u(x_1, \ldots, x_m) \leftarrow B_1 : x_1 \& \ldots \& B_n : x_m,$$

is a correct solution to fuzzy ILP task given by $\mathcal{E}$ and $\mathcal{B}$. Our presentation is here a little bit simplified and we freely switch betwee fuzzy and GAP programs, which are know to be equivalent **citace Krajci, Lencses, Vojtas FSS clanek**

III. The system prototype and Setting of our experiment

The main experiment presented in this paper leads to the seriousness classification of an accident presented on a web report. Our long term goal is extraction of semantic information from web reports. And seriousness classification is one of possible utilization of the extracted semantic information. We use web reports of fire departments of several regions of the Czech Republic. These reports are written in Czech language and can be accessed through the web of General Directorate of the Fire and Rescue Service of the Czech Republic[1]. These reports are rich in information, e.g. where and when an traffic accident occurred, which units helped, how much time it took them to show up on the place of accident, how many people were injured, killed etc.

For the present experiment we have selected a collection of 50 web reports. We have identified several features presented in these reports and manually extracted corresponding values. This will be described in more detail in section III-B. To each report we have also assigned a value of overall ranking of seriousness of presented accident, which is the target of the classification.

Máme dvě možnosti – buď odkázat na naše předchozí články o extrakci informací nebo se pokusit sem něco dát...

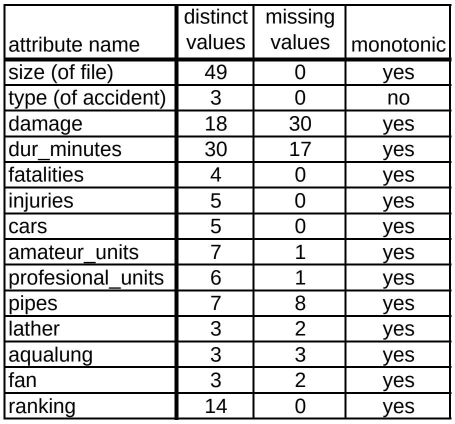
There are two objectives to do. Fist is the web information extraction, a long path starting with web crawling and resulting with the extracted structured information. Second is the seriousness classification, which utilizes the extracted information. We have made much work on the first (see e.g. [5], [6], [7]), in this paper we will concentrate on the second.

A. Experiment description

For the seriousness classification we have used two inductive logic approaches – Classical ILP and Fuzzy ILP (as described above). Technically the difference between the approaches consist in different setting of *ILP task*. Both can be done with a classical ILP tool. We have used "*A Learning Engine for Proposing Hypotheses*" (Aleph v5[2]), which seems very practical to us. It use quite effective method of *inverse entailment* [8] and keeps all handy features of Prolog system (supports YAP and SWI) in its background.

We have compared results of the two approaches (fuzzy and classical) and we could see that the fuzzy approach made better results than the classical one. See section IV for details of the results.

B. Features of accidents



attribute name	distinct values	missing values	monotonic
size (of file)	49	0	yes
type (of accident)	3	0	no
damage	18	30	yes
dur_minutes	30	17	yes
fatalities	4	0	yes
injuries	5	0	yes
cars	5	0	yes
amateur_units	7	1	yes
profesional_units	6	1	yes
pipes	7	8	yes
lather	3	2	yes
aqualung	3	3	yes
fan	3	2	yes
ranking	14	0	yes

Fig. 4. Characteristics of accident attributes.

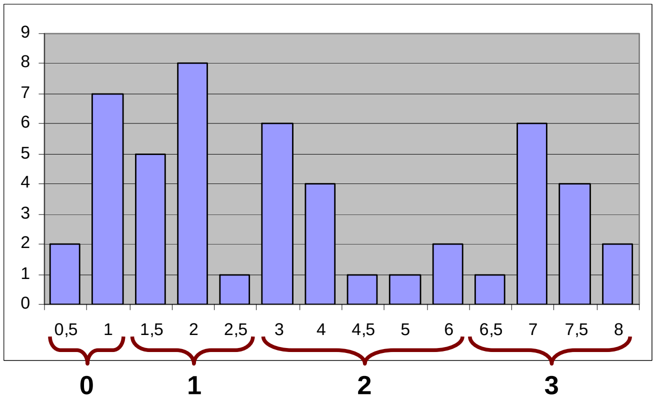


Fig. 5. Frequency histogram of accident ranking.

Figure 4 summarizes all features (or attributes) that we have obtained from accident reports. Except the attribute type (type of an accident – fire, car_accident and other) all the attributes are numerical and so monotonizable. In some cases value of some attribute is unknown. We have decided to make evidence of this and keep the values unknown in a knowledge base. Short explanation of each attribute follows.

size is a file size of text part of a report.
damage is an amount (in CZK – Czech Crowns) of summarized damage arisen during an accident.
dur_minutes is time (in minutes) taken to handle an accident.
fatalities and injuries are numbers of fatalities (and injuries) taken by an accident.
cars is number of cars damaged during an accident (especially during car accidents).
professional_units and amateur_units are numbers of fireman units sent for an accident.
pipes is number of used fire pipes.
lather, aqualung and fan (ventilator) indicates weather these devices were used.

C. Seriousness ranking

Vales of overall seriousness ranking attribute were stated from "general impression" from report's texts with respect to the particular attributes (Fig 4). Values of seriousness ranking have evolved to 14 distinct values in range form

[1] http://www.hzscr.cz
[2] http://www.comlab.ox.ac.uk/activities/machinelearning/Aleph/

		Raw ILP	Fuzzy ILP
Fuzzy test set	TP:	42	57
positive: 64	FP:	7	6
negative: 36	Precision:	0,857	0,905
sum: 100	Recall:	0,656	0,891
	F-measure:	0,743	0,898
Crisp test set	TP:	12	15
positive: 25	FP:	13	10
negative: 75	Precision:	0,480	0,600
sum: 100	Recall:	0,480	0,600
	F-measure:	0,480	0,600

Fig. 6. Evaluation results

```
serious_2(ID) :- serious_atl_2(ID),
                 not(serious_atl_3(ID)).
```

Fig. 7. Conversion of results: monotone → crisp.

0.5 to 8. Histogram with frequencies of all the values is on the Figure 5.

We have divided the values into four approximately equipotent groups (shown on the Fig. 5) and learned logic rules for each group separately.

IV. Results

See fig 6.

V. Conclusion

We have presented a proposal of and experiments with a system for semantic computing of information from Czech text on Web pages. Our system relies on linguistic annotation tools from PDT [9] and the tree querying tool Netgraph [10]. Our contributions are an experimental chain of tools that enables semantic computing. In the third phase – data extraction – we formulate an inductive logic programming task over linguistically annotated data. Finally we describe transformation of these data to an ontology. Our initial experiments verified used methods and tools.

In future work we would like to test our method on another languages and compare our results with similar solutions. Our work is very close to domain dependent information extraction such as relation and event extraction. These tasks were considered as Semantic Evaluation in the first place in the MUC-6 conference 1995 [11]. Contemporary results of

```
serious_atl_0(ID) :- serious_2(ID).
serious_atl_1(ID) :- serious_2(ID).
serious_atl_2(ID) :- serious_2(ID).
```

Fig. 8. Conversion of results: crisp → monotone.

```
damage_atl(ID,N) :- %unknown values
       damage(ID,N), not(integer(N)).
damage_atl(ID,N) :- %numeric values
       damage(ID,N2), integer(N2),
       damage(N), integer(N), N2>=N.
```

Fig. 9. Monotonization of attributes (damage → damage_atl).

```
serious_0(A):-dur_minutes(A,8).
serious_0(A):-type(A,fire), pipes(A,0).
serious_0(A):-fatalities(A,0), pipes(A,1),
              lather(A,0).
serious_1(A):-amateur_units(A,1).
serious_1(A):-amateur_units(A,0), pipes(A,2),
              aqualung(A,1).
serious_1(A):-damage(A,300000).
serious_1(A):-damage(A,unknown), type(A,fire),
              profesional_units(A,1).
serious_1(A):-dur_minutes(A,unknown),
              fatalities(A,0), cars(A,1).
serious_2(A):-lather(A,unknown).
serious_2(A):-lather(A,0), aqualung(A,1),
              fan(A,0).
serious_2(A):-amateur_units(A,2),
              profesional_units(A,2).
serious_2(A):-dur_minutes(A,unknown),
              injuries(A,2).
serious_3(A):-fatalities(A,1).
serious_3(A):-fatalities(A,2).
serious_3(A):-injuries(A,2), cars(A,2).
serious_3(A):-pipes(A,4).
```

Fig. 10. Crisp hypothesis

```
serious_atl_0(A).
serious_atl_1(A):-injuries_atl(A,1).
serious_atl_1(A):-lather_atl(A,1).
serious_atl_1(A):-pipes_atl(A,3).
serious_atl_1(A):-dur_minutes_atl(A,unknown).
serious_atl_1(A):-size_atl(A,764),
     pipes_atl(A,1).
serious_atl_1(A):-damage_atl(A,8000),
     amateur_units_atl(A,3).
serious_atl_1(A):-type(A,car_accident).
serious_atl_1(A):-pipes_atl(A,unknown),
     randomized_order_atl(A,35).
serious_atl_2(A):-pipes_atl(A,3),
     aqualung_atl(A,1).
serious_atl_2(A):-type(A,car_accident),
     cars_atl(A,2), profesional_units_atl(A,2).
serious_atl_2(A):-injuries_atl(A,1),
     profesional_units_atl(A,3), fan_atl(A,0).
serious_atl_2(A):-type(A,other),
     aqualung_atl(A,1).
serious_atl_2(A):-dur_minutes_atl(A,59),
     pipes_atl(A,3).
serious_atl_2(A):-injuries_atl(A,2),
     cars_atl(A,2).
serious_atl_2(A):-fatalities_atl(A,1).
serious_atl_3(A):-fatalities_atl(A,1).
serious_atl_3(A):-dur_minutes_atl(A,unknown),
     pipes_atl(A,3).
```

Fig. 11. Monotonised hypothesis

```
serious_2(id_47443). %positive

serious_0(id_47443). %negative
serious_1(id_47443). %negative
serious_3(id_47443). %negative
```

Fig. 12. Crisp learning examples.

```
serious_atl_0(id_47443). %positive
serious_atl_1(id_47443). %positive
serious_atl_2(id_47443). %positive

serious_atl_3(id_47443). %negative
```

Fig. 13. Monotonised learning examples.

the ACE competition[3] show the difficulty of these problems, which are very close to ours.

Acknowledgment

This work was partially supported by Czech projects IS-1ET100300517, GACR-201/09/H057 and MSM-0021620838.

References

[1] M. Mikulová, A. Bémová, J. Hajič, E. Hajičová, J. Havelka, V. Kolářová, L. Kučová, M. Lopatková, P. Pajas, J. Panevová, M. Razímová, P. Sgall, J. Štěpánek, Z. Urešová, K. Veselá, and Z. Žabokrtský, "Annotation on the tectogrammatical level in the prague dependency treebank. annotation manual," ÚFAL MFF UK, Prague, Czech Rep., Tech. Rep. 30, 2006.

[2] S. Dzeroski and N. Lavrac, Eds., *Relational Data Mining*. Berlin: Springer, 2001. [Online]. Available: http://www-ai.ijs.si/SasoDzeroski/RDMBook/

[3] S. Muggleton and L. de Raedt, "Inductive logic programming: Theory and methods," *Journal of Logic Programming*, vol. 19, pp. 629–679, 1994.

[4] T. Horváth and P. Vojtáš, "Induction of fuzzy and annotated logic programs," *Inductive Logic Programming: 16th International Conference, ILP 2006, Santiago de Compostela, Spain, August 24-27, 2006, Revised Selected Papers*, pp. 260–274, 2007.

[5] J. Dědek and P. Vojtáš, "Linguistic extraction for semantic annotation," in *2nd International Symposium on Intelligent Distributed Computing*, ser. Studies in Computational Intelligence, C. Badica, G. Mangioni, V. Carchiolo, and D. Burdescu, Eds., vol. 162. Catania, Italy: Springer-Verlag, 2008, pp. 85–94. [Online]. Available: http://www.springerlink.com/content/w7213j007t416132/

[6] ——, "Computing aggregations from linguistic web resources: a case study in czech republic sector/traffic accidents," in *Second International Conference on Advanced Engineering Computing and Applications in Sciences*, C. Dini, Ed. IEEE Computer Society, 2008, pp. 7–12. [Online]. Available: http://www2.computer.org/portal/web/csdl/doi/10.1109/ADVCOMP.2008.17

[7] J. Dědek, A. Eckhardt, and P. Vojtáš, "Experiments with czech linguistic data and ILP," in *ILP 2008 - Inductive Logic Programming (Late Breaking Papers)*, F. Železný and N. Lavrač, Eds. Prague, Czech Republic: Action M, 2008, pp. 20–25.

[8] S. Muggleton, "Inverse entailment and progol," *New Generation Computing, Special issue on Inductive Logic Programming*, vol. 13, no. 3-4, pp. 245–286, 1995. [Online]. Available: http://citeseer.ist.psu.edu/muggleton95inverse.html

[9] J. Hajič, E. Hajičová, J. Hlaváčová, V. Klimeš, J. Mírovský, P. Pajas, J. Štěpánek, B. Vidová-Hladká, and Z. Žabokrtský, "Prague dependency treebank 2.0 cd-rom," Linguistic Data Consortium LDC2006T01, Philadelphia 2006, Philadelphia, 2006.

[10] J. Mírovský, "Netgraph: A tool for searching in prague dependency treebank 2.0," in *Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories (TLT)*, J. Hajič and J. Nivre, Eds., no. 5, Prague, Czech rep., 2006, pp. 211–222.

[11] R. Grishman and B. Sundheim, "Message understanding conference-6: a brief history," in *Proceedings of the 16th conference on Computational linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 1996, pp. 466–471.

[3] `http://www.nist.gov/speech/tests/ace/`