Frequent Patterns in Natural Laguage Processing

Jan Blaťák and Luboš Popelínský

Knowledge Discovery Lab at Faculty of Informatics

Masaryk University

Botanická 68a, 602 00, Brno, Czech Republic

{xblatak,popel}@fi.muni.cz

Content

Content

- Text Mining

Content

- Text Mining
- Frequent patterns

Content

- Text Mining
- Frequent patterns
- The RAP and dRAP systems

Content

- Text Mining
- Frequent patterns
- The RAP and dRAP systems
- Data and Background Knowledge

Content

- Text Mining
- Frequent patterns
- The RAP and dRAP systems
- Data and Background Knowledge
- Frequent patterns in Text Mining

Content

- Text Mining
- Frequent patterns
- The RAP and dRAP systems
- Data and Background Knowledge
- Frequent patterns in Text Mining
- Context-Sensitive Text Correction

Content

- Text Mining
- Frequent patterns
- The RAP and dRAP systems
- Data and Background Knowledge
- Frequent patterns in Text Mining
- Context-Sensitive Text Correction
- Morphological Disambiguation of Czech

Content

- Text Mining
- Frequent patterns
- The RAP and dRAP systems
- Data and Background Knowledge
- Frequent patterns in Text Mining
- Context-Sensitive Text Correction
- Morphological Disambiguation of Czech
- Information Extraction LLL05 Challenge

Content

- Text Mining
- Frequent patterns
- The RAP and dRAP systems
- Data and Background Knowledge
- Frequent patterns in Text Mining
- Context-Sensitive Text Correction
- Morphological Disambiguation of Czech
- Information Extraction LLL05 Challenge
- Summary and Conclusions

Motivation: Text Mining

Text mining

- *text classification*, *information extraction*, *summarization*, *disambiguation (morphological, word-sense, . . .)* etc.

Two usual approaches

1. *Ad hoc* data transformation (preprocessing) + attribute-value learners (Naïve Bayes, SVM, . . .)
   – appropriate for text classification
   – difficult to incorporate additional information (morphology, etc.)
2. Relational Data Mining (an ILP system + specialized background knowledge)
   – easily extensible
   – appropriate for complex data (morpho-syntactic relations, etc.)

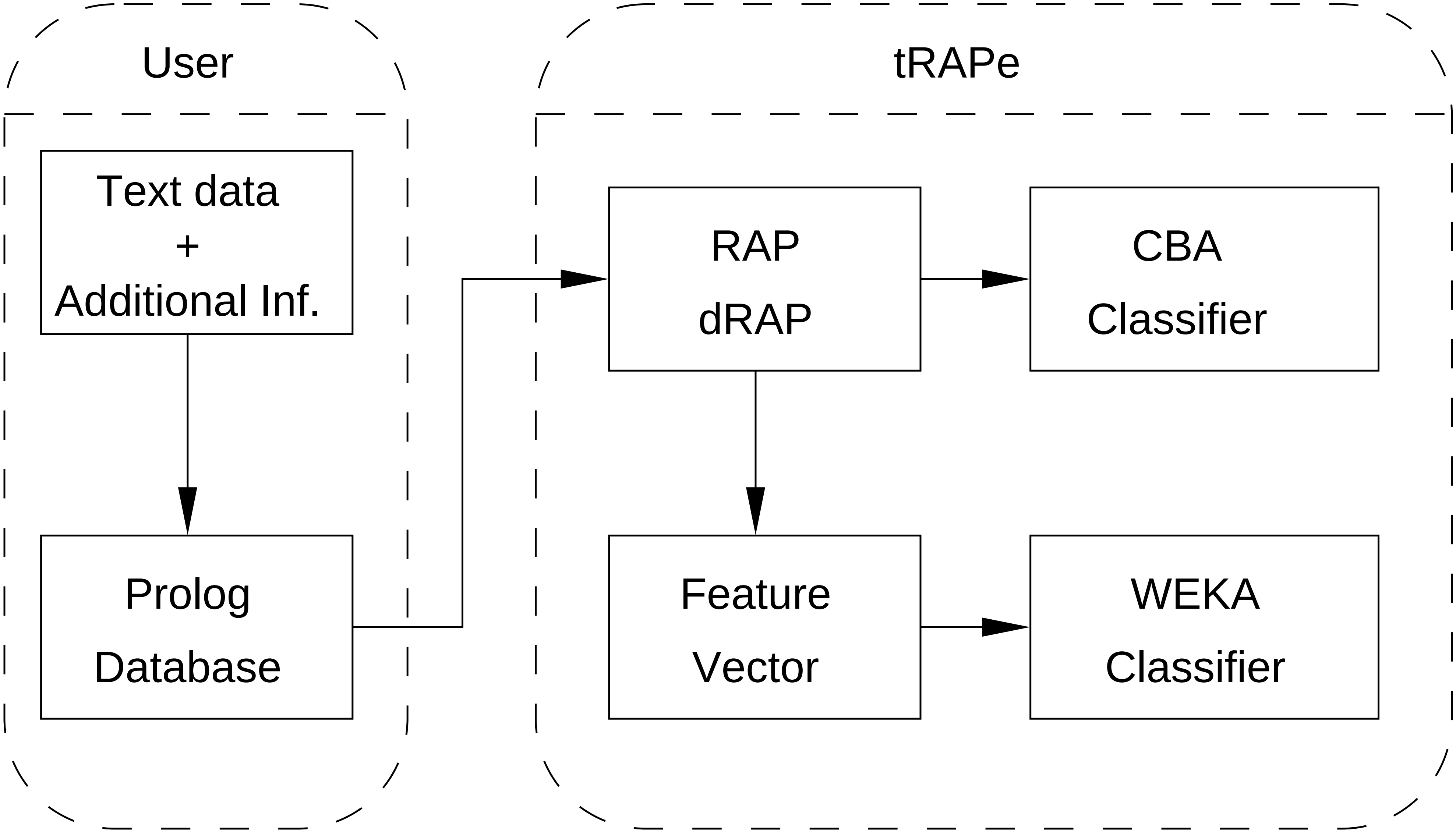
Drawbacks

- No general method or system exists

Goals

1. To design a general framework for solving text mining tasks by using long first-order frequent patterns
2. To use frequent patterns as new features (propositionalization) or to construct class-based association rules (CAR)
3. To evaluate this framework on real world datasets and tasks

Motivation: Text Mining Process



Frequent patterns

F – minimal frequency threshold given by the user

Frequent pattern

A conjunction of literals which covers at least F examples

Maximal frequent pattern

A frequent pattern whose extensions are not frequent patterns

Algorithms for finding frequent patterns

- *Propositional data:* the *Apriori* algorithm [Agrawal and Srikant, 1994]
- *First-order logic:* the WARMR level-wise system [Dehaspe & Toivonen, 1999]
- *Maximal first-order frequent patterns:* the RAP system [Blaťák *et al.*, 2002]

Frequent patterns: Example

Let us have a database of right contexts of the word `among`

```
among several sovereign states has ...
among other matters, investigating ...
among the young who have ...
among the top three places ...
among scattershot releases. ...
```

and a minimal frequency threshold $F = 2$

Frequent patterns: Example

Let us have a database of right contexts of the word among and a minimal frequency threshold $F = 2$

```
1. [among/IN] several/JJ sovereign/JJ states/NNS has/VBZ
2. [among/IN] other/JJ matters/NNS ,/, investigating/VBG
3. [among/IN] the/DT young/JJ who/WP have/VBP
4. [among/IN] the/DT top/JJ three/CD  places/NNS
5. [among/IN] scattershot/JJ releases/NNS ./.
```

Frequent patterns: Example

Let us have a database of right contexts of the word among and a minimal frequency threshold $F = 2$

```
1. [among/IN] several/JJ sovereign/JJ states/NNS has/VBZ
2. [among/IN] other/JJ matters/NNS ,/, investigating/VBG
3. [among/IN] the/DT young/JJ who/WP have/VBP
4. [among/IN] the/DT top/JJ three/CD  places/NNS
5. [among/IN] scattershot/JJ releases/NNS ./.
```

Propositional patterns

- word/tag – just one pattern – the/DT [supp. 2]
- tag – DT [2], JJ [5], NNS [4], DT & JJ [2], JJ & NNS [3]

Frequent patterns: Example

Let us have a database of right contexts of the word among and a minimal frequency threshold *F* = 2

```
1. [among/IN] several/JJ sovereign/JJ states/NNS has/VBZ
2. [among/IN] other/JJ matters/NNS ,/, investigating/VBG
3. [among/IN] the/DT young/JJ who/WP have/VBP
4. [among/IN] the/DT top/JJ three/CD  places/NNS
5. [among/IN] scattershot/JJ releases/NNS ./.
```

Propositional patterns

- word/tag – just one pattern – the/DT [supp. 2]
- tag – DT [2], JJ [5], NNS [4], DT & JJ [2], JJ & NNS [3]

First-order patterns

- word/tag – *hasToken(X), is-a(X,'the/DT').* [2]
- tag – *hasToken(X), is-a(X,'JJ'), follows(Y,X), is-a(Y,'NNS').* [3/2]
- word + tag – *hasToken(X), is-a(X,'the'), follows(Y,X), is-a(Y,'JJ').* [2]
- meta – *hasToken(X), is-a(X,punctuation).* [2]

RAP [Blaťák & Popelínský, 2004]

A system for mining long (maximal) first-order frequent patterns

Features

- Intended for mining “interesting” patterns from dense data
  - best-first search + strong pruning
  - depth-first and random search are also implemented
- Any-time algorithm
  - it is possible to generate all frequent patterns

RAP [Blaťák & Popelínský, 2004]

A system for mining long (maximal) first-order frequent patterns

Features

- Intended for mining “interesting” patterns from dense data
  - best-first search + strong pruning
  - depth-first and random search are also implemented
- Any-time algorithm
  - it is possible to generate all frequent patterns

Why long frequent patterns?

RAP [Blaťák & Popelínský, 2004]

A system for mining long (maximal) first-order frequent patterns

Features

- Intended for mining “interesting” patterns from dense data
  – best-first search + strong pruning
  – depth-first and random search are also implemented
- Any-time algorithm
  – it is possible to generate all frequent patterns

Why long frequent patterns?

- Short patterns are usually too general (redundant – cover the same examples)
- Long patterns are usually better for revealing long-distance dependencies [Cussens, 1997; Nepil, 2003]
- Minimal frequency threshold prevents system from overfitting

dRAP [Blaťák, 2005]

Problems of ILP systems

- Impossible to process a large scale of data (usual solutions – splitting the data [Cussens *et al.*, 2000], selective sampling [Nepil, 2003])
- Time consuming theory evaluation (54.5 hours for learning disambiguation rules for pronoun in Slovene [Cussens *et al.*, 2000], three days for whole theory [Nepil, 2003])

dRAP [Blaťák, 2005]

Problems of ILP systems

- Impossible to process a large scale of data (usual solutions – splitting the data [Cussens *et al.*, 2000], selective sampling [Nepil, 2003])
- Time consuming theory evaluation (54.5 hours for learning disambiguation rules for pronoun in Slovene [Cussens *et al.*, 2000], three days for whole theory [Nepil, 2003])

dRAP: *An extension of the RAP system designed for mining in distributed data*

dRAP [Blaťák, 2005]

Problems of ILP systems

- Impossible to process a large scale of data (usual solutions – splitting the data [Cussens *et al.*, 2000], selective sampling [Nepil, 2003])
- Time consuming theory evaluation (54.5 hours for learning disambiguation rules for pronoun in Slovene [Cussens *et al.*, 2000], three days for whole theory [Nepil, 2003])

dRAP: *An extension of the RAP system designed for mining in distributed data*

- Distributed data algorithm (based on Savasere's approach [Savasere *et al.*, 1995])
- No communication between the computational nodes
- Master-worker architecture
- Two phase computation
  - generation of locally frequent maximal patterns (workers)
  - merging locally frequent patterns (master)

Data

Flat data representation: *w(SiD, WiD, Word).*

Data

Flat data representation: *w(SiD, WiD, Word).*

SiD – sentence identifier

WiD – word position in the sentence

Word – word (string, i.e., list of character codes)

Data

Flat data representation: *w(SiD, WiD, Word).*

SiD – sentence identifier

WiD – word position in the sentence

Word – word (string, i.e., list of character codes)

Example: “. . . to be shared by those among its 600 computer experts who . . . ”

Data

Flat data representation: *w(SiD, WiD, Word).*

SiD – sentence identifier

WiD – word position in the sentence

Word – word (string, i.e., list of character codes)

Example: "... to be shared by those among its 600 computer experts who ..."

- words (required):

```
w(a1DW,16,"to").   w(a1DW,17,"be").   w(a1DW,18,"shared").   ...
```

Data

Flat data representation: *w(SiD, WiD, Word).*

SiD – sentence identifier

WiD – word position in the sentence

Word – word (string, i.e., list of character codes)

Example: "...to be shared by those among its 600 computer experts who..."

- words (required):

  `w(a1DW,16,"to").   w(a1DW,17,"be").   w(a1DW,18,"shared").   ...`

- lemma:

  `l(a1DW,16,"to").   l(a1DW,17,"be").   l(a1DW,18,"share").   ...`

Data

Flat data representation: *w(SiD, WiD, Word).*

SiD – sentence identifier

WiD – word position in the sentence

Word – word (string, i.e., list of character codes)

Example: "...to be shared by those among its 600 computer experts who..."

- words (required):

```
w(a1DW,16,"to").   w(a1DW,17,"be").   w(a1DW,18,"shared").   ...
```

- lemma:

```
l(a1DW,16,"to").   l(a1DW,17,"be").   l(a1DW,18,"share").   ...
```

- part-of-speech:

```
t(a1DW,16,"TO").   t(a1DW,17,"VB").   t(a1DW,18,"VBN").   ...
```

Data

data generated automatically

from arbitrary plain text

and/or from the output of Memory-based Shallow Parser (Daelmans et al.)

Background Knowledge

Background Knowledge

Common predicates

focusWord/2 – introduces the focus word

begCap/2 – the first letter of a given word is capital

isPunct/2 and *isQuot/2* – given token is a special character

Background Knowledge

Common predicates

focusWord/2 – introduces the focus word

begCap/2 – the first letter of a given word is capital

isPunct/2 and *isQuot/2* – given token is a special character

Structural predicate in $\mathcal{B}^1$

hasWord/3 – introduces a word from relative position given in the argument

Background Knowledge

Common predicates

focusWord/2 – introduces the focus word

begCap/2 – the first letter of a given word is capital

isPunct/2 and *isQuot/2* – given token is a special character

Structural predicate in $\mathcal{B}^1$

hasWord/3 – introduces a word from relative position given in the argument

Structural predicates in $\mathcal{B}^2$

leftWord/2 – introduces some word from left context

rightWord/2 – introduces some word from right context

Background Knowledge: Examples

Background knowledge $\mathcal{B}^1$

"...World Cup semifinal between England and Germany in 1990..."

focusWord(A,B), hasWord(1,B,C), begCap(A,C), hasWord(2,B,D)

Background knowledge $\mathcal{B}^2$

"...time that relations between the United States and China..."

focusWord(A,B), rightWord(B,C), begCap(A,C), rightWord(C,D)

Background Knowledge: Examples

Background knowledge $\mathcal{B}^1$

"... World Cup semifinal between England and Germany in 1990 ..."

focusWord(A,B), hasWord(1,B,C), begCap(A,C), hasWord(2,B,D)

Background knowledge $\mathcal{B}^2$

"... time that relations between the United States and China ..."

focusWord(A,B), rightWord(B,C), begCap(A,C), rightWord(C,D)

Background Knowledge: Examples

Background knowledge $\mathcal{B}^1$

"…World Cup semifinal between England and Germany in 1990…"

focusWord(A,B), hasWord(1,B,C), begCap(A,C), hasWord(2,B,D)

Background knowledge $\mathcal{B}^2$

"…time that relations between the United States and China…"

focusWord(A,B), rightWord(B,C), begCap(A,C), rightWord(C,D)

Background Knowledge: Examples

Background knowledge $\mathcal{B}^1$

"... World Cup semifinal between England and Germany in 1990 ..."

focusWord(A,B), hasWord(1,B,C), begCap(A,C), hasWord(2,B,D)

Background knowledge $\mathcal{B}^2$

"... time that relations between the United States and China ..."

focusWord(A,B), rightWord(B,C), begCap(A,C), rightWord(C,D)

Background Knowledge: Examples

Background knowledge $\mathcal{B}^1$

"...World Cup semifinal between England and Germany in 1990..."

focusWord(A,B), hasWord(1,B,C), begCap(A,C), hasWord(2,B,D)

Background knowledge $\mathcal{B}^2$

"...time that relations between the United States and China..."

focusWord(A,B), rightWord(B,C), begCap(A,C), rightWord(C,D)

Feature Construction (Propositionalization)

Propositionalization [Kramer *et al.*, 2001]

- *A process in which a relational data are transformed into an attribute-value (propositional) form*

Feature

- Defined as a rule of the form $f_i(X) : -Lit_{i,1}, \ldots, Lit_{i,n_i}$
  - $Lit_{i,k}$ ($k \in \mathbb{N}$) is a literal from background knowledge
  - X is an example identifier

In this approach

- body of the rule is a frequent pattern

Feature-vector

- Fixed size vector: $f_1(X) = v_1 \wedge f_2(X) = v_2 \wedge \ldots \wedge f_m(X) = v_m$

  where $v_i = 1$ if $f_i(X)$ holds, $v_i = 0$ otherwise

CBA: Class Based Association

Class association rule (CAR) [Liu *et al.*, 1998]

- *An association rule which has only a class identifier in the consequent (head)*

CBA classifier [Liu *et al.*, 1998]

- *A collection of class association rules*

Classification with CBA

- *By majority*: most frequent class is assigned
- *Sequential classification*: for a given ordering of classes, a class c is assigned if
  1. a CAR Q for class c covers example and covers at least *MinCov* examples from class c in training data
  2. at least *MinNum* rules for class c cover example

Experiments: Environment & Settings

Environment

- AMD Atlon^{TM} XP 2500+ with 756 MB of memory
- Linux Fedora^{TM} Core 3
- *Distributed mining:* four nodes
- *Classification:* SMO (SVM), J48 (IDT), Naïve Bayes and IB1 (Instance Based) learners from the Weka package [Witten & Frank, 1999]

Settings

- The Background knowledge $\mathcal{B}_1$ or $\mathcal{B}_2$ + task specific predicates
- Minimal frequency threshold between 1 and 10 %

Context-sensitive text correction

Motivation

- *Problem:* Current spell checkers do not use context information:
  - "*I'd like a peace of cake.*" instead of "*I'd like a piece of cake.*"

Context-sensitive text correction

Motivation

- *Problem:* Current spell checkers do not use context information:
  - "*I'd like a peace of cake.*" instead of "*I'd like a piece of cake.*"
- *Solution:* To use context for determining correct word [Carlson *et al.*, 2001]

Context-sensitive text correction

Motivation

- *Problem:* Current spell checkers do not use context information:
  - "*I'd like a peace of cake.*" instead of "*I'd like a piece of cake.*"
- *Solution:* To use context for determining correct word [Carlson *et al.*, 2001]

Task definition

- To generate rules for words *among* and *between*

Context-sensitive text correction

Motivation

- *Problem:* Current spell checkers do not use context information:
  - "*I'd like a peace of cake.*" instead of "*I'd like a piece of cake.*"
- *Solution:* To use context for determining correct word [Carlson *et al.*, 2001]

Task definition

- To generate rules for words *among* and *between*

Data

- TDT2 English corpus [Carlson *et al.*, 2001]
- Additional information: morphology (SNoW-based part-of-speech tagger)
- Number of occurrences: 7,119 for *among* and 13,378 for *between*
- Training data: 16,398 contexts of the length five words

Context-sensitive text correction: Rule Examples

Background knowledge: $\mathcal{B}^1$

key(A), focusWord(A,B), hasWord(1,B,C), begCap(A,C), hasTag(A,C,'NNP'), hasWord(2,B,D), hasTag(A,D,'CC').

- "…*semifinal/NNP* [between/IN]^B *England/NNP*^{C} *and/CC*^{D} *Germany/NNP in/IN 1990/CD*…"
- Class distribution: among – 16, between – 1397 [supp. 1413]
- Precision: 98.85 %

Context-sensitive text correction: Rule Examples

Background knowledge: $\mathcal{B}^1$

key(A), focusWord(A,B), hasWord(1,B,C), begCap(A,C), hasTag(A,C,'NNP'), hasWord(2,B,D), hasTag(A,D,'CC').

- "...*semifinal/NNP* [between/IN]^B *England/NNP*^{C} *and/CC*^{D} *Germany/NNP in/IN 1990/CD*..."
- Class distribution: among – 16, between – 1397 [supp. 1413]
- Precision: 98.85 %

Background knowledge: $\mathcal{B}^2$

key(A), focusWord(A,B), rightWord(B,C), begCap(A,C), hasTag(A,C,'NNP'), rightWord(C,D), hasTag(A,D,'CC').

- "...*relations/NNS* [between/IN]^B *the/DT United/NNP*^{C} *States/NNP*$^{\text{another } C}$ *and/CC*^{D} *China/NNP*..."
- Class distribution: among – 92, between – 3095 [supp. 3187]
- Precision: 97.11 %

Context-sensitive text correction: Propositionalization

		RAP				dRAP			
		$\mathcal{B}_{\mathcal{S}}^1$		$\mathcal{B}_{\mathcal{S}}^2$		$\mathcal{B}_{\mathcal{S}}^1$		$\mathcal{B}_{\mathcal{S}}^2$	
Cls.	IW	Prec./Rec./F_1	Acc.	Prec./Rec./F_1	Acc.	Prec./Rec./F_1	Acc.	Prec./Rec./F_1	Acc.
SVM	am.	.61/.80/.69	.75	.69/.80/.74	**.80**	.69/.71/.70	.79	.70/.78/.73	**.80**
	bet.	.87/.72/.79		.88/.81/.84		.84/.83/.84		.87/.82/.85	
J48	am.	.63/.75/.68	.76	.69/.80/.74	.80	.72/.73/.73	**.81**	.71/.77/.74	**.81**
	bet.	.85/.76/.80		.88/.81/.84		.86/.85/.85		.87/.83/.85	
NB	am.	.60/.78/.68	.74	.62/.90/.73	**.77**	.58/.81/.67	.73	.58/.83/.69	.74
	bet.	.86/.72/.78		.93/.70/.80		.87/.68/.77		.88/.68/.77	

IW – intended word (am. – among, bet – between)

Acc. – accuracy (baseline = 62.5 %)

running time of dRAP = 1/num_of_nodes * time_of_RAP

Context-sensitive text correction:

Minimal frequency thresholds: 10 %

Background knowledge: $\mathcal{B}^1$

CBA method: by majority

		RAP			dRAP		
Rules	IW	#	Prec./Rec./F_1	Acc.	#	Prec./Rec./F_1	Acc.
max	am.	0	–	.65	9	.62/.50/.55	**.72**
	bet.	18	.71/.86/.78		31	.76/.83/.80	
freq	am.	0	–	.56	12	.59/.64/.62	**.71**
	bet.	24	.65/1.0/.79		38	.80/.76/.78	

Rules – used rules (max – only maximal frequent patterns, freq – all frequent patterns)

– number of class association rules

Morphological Disambiguation of Czech

Czech morphology

- Czech is highly inflectional Slavic language
- Many possible morpho-syntactical readings for each word

Task definition

- To recognize the correct morphological reading of the word “je” [Popelínský & Pavelek, 1999]
  - Pronoun *them* (e.g. “I see *them*.”)
  - Verb *to be/is* (e.g. “He *is* a driver.”)
  - *Interjection* (it is too rare)

Learning set:

- DESAM [Pala *et al.*, 1997], an annotated corpus for Czech.
- Number of occurrences: 9360 *(verb)*, 703 *(pronoun)*

Morphological Disambiguation of Czech: Data Example

Přihlášku	přihláška	k1gFnSc4,	application form
je	být	k5mIp3nSaI	is
je	on	k3p3gMnPc4, k3p3gInPc4, k3p3gNnSc4, k3p3gNnPc4, k3p3gFnPc4	them
je	je	k0	
třeba	třeba	k6xDd1 k8 k9	neccessary
podat	podat	k5mFaP	admit
nejpozději	pozdě	k6xMd3	late
do	do	k7	to
konce	konec	k1gInSc2, k1gInPc1, k1gInPc4, k1gInPc5	end
dubna	duben	k1gInSc2	April
.	.	kI	

Morphological Disambiguation of Czech: Data & Settings

RAP

- Number of examples: 100 (50 for each class)

dRAP

- Number of examples: 400 (200 for each class)
- Relaxed pruning

Background knowledge

- Type: $\mathcal{B}^1$
- Additional predicate: *hasTag* for introducing
  - part-of-speech, case, gender, number,...

Testing

- Testing on 600 unseen examples (300 for each class)

Morphological Disambiguation of Czech: Propositionalization

Rules: all frequent patterns

		RAP		dRAP	
Cls.	Sense	Prec./Rec./F_1	Acc.	Prec./Rec./F_1	Acc.
SVM	pronoun	.85/.64/.73	76.0%	.80/.89/**.84**	***83.5%***
	verb	.71/.88/.79		.88/.78/**.83**	
J48	pronoun	.98/.47/.64	73.0%	.68/.90/**.77**	**73.8%**
	verb	.65/.99/**.79**		.85/.58/.69	
NB	pronoun	.76/.80/.78	77.5%	.86/.79/**.82**	**83.0%**
	verb	.79/.75/.77		.81/.87/**.84**	

Sense – intended morphological meaning

Morphological Disambiguation of Czech: CBA

Rules: all frequent patterns

CBA method: by majority

	RAP			dRAP		
Sense	#	Prec./Rec./F_1	Acc.	#	Prec./Rec./F_1	Acc.
k3	20	78/.75/.77	71.2%	35	.86/.71/.77	***76.7%***
k5	19	.82/.68/.74		36	.79/.83/**.81**	

Sense – intended morphological meaning (k3 – pronoun “*them*”, k5 – verb “*is*”)

– number of class association rules

LLL05: Information Extraction

“*GerE stimulates cotD transcription and inhibits cotA transcription in vitro by sigma K RNA polymerase, as expected from in vivo studies, and, unexpectedly, profoundly inhibits in vitro transcription of the gene (sigK) that encode sigma K.*”

Biological texts

LLL05: Information Extraction

“*GerE stimulates cotD transcription and inhibits cotA transcription in vitro by sigma K RNA polymerase, as expected from in vivo studies, and, unexpectedly, profoundly inhibits in vitro transcription of the gene (sigK) that encode sigma K.*”

Biological texts

- Goal is to determine gen-protein interactions

LLL05: Information Extraction

"*GerE stimulates cotD transcription and inhibits cotA transcription in vitro by sigma K RNA polymerase, as expected from in vivo studies, and, unexpectedly, profoundly inhibits in vitro transcription of the gene (sigK) that encode sigma K.*"

Biological texts

- Goal is to determine gen-protein interactions

  GerE-cotD, GerE-cotA, sigma K-cotA, GerE-SigK and sigK-sigma K

LLL05: Information Extraction

"*GerE stimulates cotD transcription and inhibits cotA transcription in vitro by sigma K RNA polymerase, as expected from in vivo studies, and, unexpectedly, profoundly inhibits in vitro transcription of the gene (sigK) that encode sigma K.*"

Biological texts

- Goal is to determine gen-protein interactions

  GerE-cotD, GerE-cotA, sigma K-cotA, GerE-SigK and sigK-sigma K

- Interactions described with natural language

LLL05: Information Extraction

"GerE stimulates cotD transcription and inhibits cotA transcription in vitro by sigma K RNA polymerase, as expected from in vivo studies, and, unexpectedly, profoundly inhibits in vitro transcription of the gene (sigK) that encode sigma K."

Biological texts

- Goal is to determine gen-protein interactions

  GerE-cotD, GerE-cotA, sigma K-cotA, GerE-SigK and sigK-sigma K

- Interactions described with natural language

- Additional information is available: morpho-syntactic relations, lemmas

LLL05: Information Extraction

"GerE stimulates cotD transcription and inhibits cotA transcription in vitro by sigma K RNA polymerase, as expected from in vivo studies, and, unexpectedly, profoundly inhibits in vitro transcription of the gene (sigK) that encode sigma K."

Biological texts

- Goal is to determine gen-protein interactions

  GerE-cotD, GerE-cotA, sigma K-cotA, GerE-SigK and sigK-sigma K

- Interactions described with natural language
- Additional information is available: morpho-syntactic relations, lemmas
- Data intended for relational mining

LLL05: Data characteristics

LLL05: Data characteristics

Training data

- Tokens/words and their position in a sentence
- Lemmas
- Morphology + morpho-syntactic relations
- List of agents and goals
- Number of interactions: 103 positive + 473 negative = 576 (testing on another 660 interactions)

LLL05: Data characteristics

Training data

- Tokens/words and their position in a sentence
- Lemmas
- Morphology + morpho-syntactic relations
- List of agents and goals
- Number of interactions: 103 positive + 473 negative = 576 (testing on another 660 interactions)

Background knowledge

- Type: $\mathcal{B}^2$

LLL05: Data characteristics

Training data

- Tokens/words and their position in a sentence
- Lemmas
- Morphology + morpho-syntactic relations
- List of agents and goals
- Number of interactions: 103 positive + 473 negative = 576 (testing on another 660 interactions)

Background knowledge

- Type: $\mathcal{B}^2$

Additional predicates:

- Morphology: *noun/2*, *verb/2*, . . .
- Morpho-syntactic relations: *isObj/2*, *isSubj/2*, . . .

LLL05: Propositionalization

Measures: *Precision/Recall/*F_1 *measure*

	RAP			dRAP		
Cls.	D$_M$	D$_F$	D$_E$	D$_M$	D$_F$	D$_E$
SVM	–	–	.34/.30/.32	–	–	.31/.39/**.35**
J48	–	–	.35/.20/.26	.36/.07/.12	.50/.03/.07	.33/.22/**.27**
NB	.14/.06/.08	.14/.15/.14	.18/.18/.18	.17/.13/.15	.27/.24/.25	.34/.26/**.29**
IB1	.20/.19/.19	.21/.26/.23	.19/.26/.22	.40/.33/**.36**	.11/.17/.14	.24/.31/.27

D$_M$ – maximal frequent patterns

D$_F$ – all frequent patterns

D$_E$ – all patterns which cover at least one example

'–' – all examples classified into the majority class

LLL05: CBA Classification

Measures: *Precision/Recall/*F_1 *measure*

Rules: all rules which cover at least one example (interaction)

CBA method: sequential classification

		RAP		dRAP	
T_H^+	T_H^-	Dis$^-$	Dis$^+$	Dis$^-$	Dis$^+$
4/2	3/2	.32/.20/.25	.17/.20/.19	.36/.28/**.31**	.21/.28/.24
5/3	3/2	.35/.11/.17	.12/.11/.11	.48/.19/**.27**	.19/.19/.19

T_H^+ & T_H^- – the value of thresholds *MinCov/MinNum* (positive interaction & negative interaction)

Dis$^+$ – both, negative and positive rules were used

Dis$^-$ – only positive rules were used

Summary

Summary

- Three different tasks were solved

Summary

- Three different tasks were solved
- For all the tasks propositionalization performed better than the CBA classifier
- The background knowledge $\mathcal{B}^2$ provides better results than $\mathcal{B}^1$

Summary

- Three different tasks were solved
- For all the tasks propositionalization performed better than the CBA classifier
- The background knowledge $\mathcal{B}^2$ provides better results than $\mathcal{B}^1$
- For all tasks large number of generated features means better results despite of feature overlapping

Current & Future Work

Current & Future Work

- Data extended with an output from a shallow parser

Current & Future Work

- Data extended with an output from a shallow parser
- This data have been exploited for mining news reports on flood (situation and action discovery)

Current & Future Work

- Data extended with an output from a shallow parser
- This data have been exploited for mining news reports on flood (situation and action discovery)
- First version of a refinement for spatio-temporal data added

Current & Future Work

- Data extended with an output from a shallow parser
- This data have been exploited for mining news reports on flood (situation and action discovery)
- First version of a refinement for spatio-temporal data added

Current & Future Work

- Data extended with an output from a shallow parser
- This data have been exploited for mining news reports on flood (situation and action discovery)
- First version of a refinement for spatio-temporal data added
- Automatic method for tuning parameters like a minimum frequency, *MinCov* and *MinNum* would help

Thank you for attention