University of Zagreb
Faculty of Electrical Engineering and Computing
Text Analysis and Knowledge Engineering Lab

Guessing the Correct Inflectional Paradigm of Unknown Croatian Words

Jan Šnajder

Eighth Language Technologies Conference (IS-JT'12)
Ljubljana, October 8th, 2012

UNKNOWN

. . . koji je vrijeđate svojim **nelajkanjem** pa makar . . .

Motivation

- A real-life morphological analyzer must be able to handle **out-of-vocabulary words**
- Analyzers for inflectionally rich languages typically rely on **morphological lexica**
- Lexica are inevitably of **limited coverage**
- Solution is to use a **morphological guesser** to determine the unknown word's stem, tags, paradigm/pattern, etc.
- Useful for:
  - lexicon acquisition/enlargement
  - morphological tagging

Motivation

take[lab];

- A real-life morphological analyzer must be able to handle **out-of-vocabulary words**
- Analyzers for inflectionally rich languages typically rely on **morphological lexica**
- Lexica are inevitably of **limited coverage**
- Solution is to use a **morphological guesser** to determine the unknown word's stem, tags, paradigm/pattern, etc.
- Useful for:
  - lexicon acquisition/enlargement
  - morphological tagging

Our aim

take[lab];

- Guess the **inflectional paradigm** (and lemma) of unknown Croatian words
  1. use a **morphological grammar** to generate candidate lemma-paradigm pairs
  2. use **supervised machine learning** to train a model to decide which pair is correct based on a number of features
- We focus on machine learning aspects: what are the relevant features and how well can we do?

Our aim

take[lab];

- Guess the **inflectional paradigm** (and lemma) of unknown Croatian words
  1. use a **morphological grammar** to generate candidate lemma-paradigm pairs
  2. use **supervised machine learning** to train a model to decide which pair is correct based on a number of features
- We focus on machine learning aspects: what are the relevant features and how well can we do?

Outline

take[lab];

- Problem definition
- Features
- Evaluation
- Remarks
- Conclusion

Problem definition

- Given word-form w, determine its correct stem s and its correct inflectional paradigm p
- Given p, the lemma l can be derived from the stem s and vice versa, thus the problem can be re-casted as:

Problem definition

Given word-form w, determine its correct lemma-paradigm pair (LPP) (l, p).
LPP is correct iff l is valid and p generates the valid word-forms of the stem obtained from l.

- E.g. for $w =$ *nelajkanjem*:
  (*nelajkanje*, $N28$) is correct, but (*nelajkanj*, $A06$) isn't

Problem definition

- Given word-form w, determine its correct stem s and its correct inflectional paradigm p
- Given p, the lemma l can be derived from the stem s and vice versa, thus the problem can be re-casted as:

Problem definition

Given word-form w, determine its correct lemma-paradigm pair (LPP) (l, p).
LPP is correct iff l is valid and p generates the valid word-forms of the stem obtained from l.

- E.g. for $w =$ *nelajkanjem*:
  $($*nelajkanje*$, N28)$ is correct, but $($*nelajkanj*$, A06)$ isn't

LPP generation

take[lab];

- First step is candidate LPP generation using a **morphology grammar**
- Grammar must be generative and reductive
- We use the **Croatian HOFM grammar** (Šnajder & Dalbelo Bašić 2008; Šnajder 2010)
- **93 paradigms**: 48 for nouns, 13 for adjectives, 32 for verbs
- Uses MULTEXT-East morphological tags (Erjavec 2003)
- Grammar is ambiguous: on average, each word-form is lemmatized to **17 candidate LPPs**

Morphology grammar – example

take[lab];

Word-form generation

```
> wfs "vojnik" N04
[("vojnik","N-msn"),("vojnika","N-msg"),
 ("vojnika","N-msa"),("vojnika","N-mpg"),
 ("vojniku","N-msl"),("vojniče","N-msv"),...]
```

Word-form lemmatization

```
> lm "vojnika"
[("vojnik",N01),("vojnikin",N03),
 ("vojnik",N04),("vojniak",N05),
 ("vojniak",N06),("vojniko",N17),...]
```

LPP classification

take[lab];

- Binary classification problem (which candidate LPP is correct?)
- SVM with RBF kernel (#features $\ll$ #examples)
- Training/testing data: semi-automatically acquired inflectional lexicon from (Šnajder 2008) with 68,465 LPPs

Features

take[lab];

- **String-based features** – orthographic properties of lemma/stem
  - incorrect LPPs tend to generate ill-formed stems/lemmas
- **Corpus-based features** – frequencies or probability distributions of word-forms/morphological tags in the corpus
  - a correct LPP should have more of its word-forms attested in the corpus
  - every inflectional paradigm has its own distribution of morphological tags $P(t|p)$. A correct LPP will generate word-forms that obey such a distribution
- **Other features** – *paradigmId* and *POS*
- **22 features** in total (146 binary-encoded)

Features

- **String-based features** – orthographic properties of lemma/stem
  - incorrect LPPs tend to generate ill-formed stems/lemmas
- **Corpus-based features** – frequencies or probability distributions of word-forms/morphological tags in the corpus
  - a correct LPP should have more of its word-forms attested in the corpus
  - every inflectional paradigm has its own distribution of morphological tags $P(t|p)$. A correct LPP will generate word-forms that obey such a distribution
- **Other features** – *paradigmId* and *POS*
- **22 features** in total (146 binary-encoded)

Features

- **String-based features** – orthographic properties of lemma/stem
  - incorrect LPPs tend to generate ill-formed stems/lemmas
- **Corpus-based features** – frequencies or probability distributions of word-forms/morphological tags in the corpus
  - a correct LPP should have more of its word-forms attested in the corpus
  - every inflectional paradigm has its own distribution of morphological tags $P(t|p)$. A correct LPP will generate word-forms that obey such a distribution
- **Other features** – *paradigmId* and *POS*
- **22 features** in total (146 binary-encoded)

String-based features

take[lab];

1. *EndsIn*
2. *EndsInCgr*
3. *EndsInCons*
4. *EndsInNonPals*
5. *EndsInPals*
6. *EndsInVelars*
7. *LemmaSuffixProb* – the probability $P(s_l|p)$
8. *StemSuffixProb* – the probability $P(s_s|p)$
9. *StemLength*
10. *NumSyllables*
11. *OneSyllable*

Corpus-based features

take[lab];

1. *LemmaAttested*
2. *Score0* – number of attested word-form types
3. *Score1* – sum of corpus frequencies of word-forms
4. *Score2* – proportion of attested word-form types
5. *Score3* – product of $P(t|p)$ and $P(t|l,p)$
6. *Score4* – expected number of attested word-form types
7. *Score5* – Kullback-Leibler divergence between $p_1 = P(t|p)$ and $p_2(t) = P(t|l,p)$
8. *Score6* – Jensen-Shannon divergence between p_1 and p_2
9. *Score7* – cosine similarity between p_1 and p_2

- Estimated on *Vjesnik* newspaper corpus (23 MW)

Evaluation – data set

- **Positive examples**: LPPs sampled from the lexicon – 5,000 for training and 5,000 for testing
- **Negative examples**: generated using the grammar – 5,000 for training and 5,000 for testing
- Total: 10,000 examples for training and 10,000 examples for testing
- Ought to be sufficient (146 features vs. 10,000 examples)

Evaluation – feature analysis

take[lab];

- Some features are **redundant** while others may be **irrelevant**
- Top-5 features with **univariate filter selection**:
  - IG: *StemSuffixProb*, *LemmaSuffixProb*, *Score6*, *Score5*, *Score7*
  - GR: *StemSuffixProb*, *LemmaSuffixProb*, *LemmaAttested*, *Score0*, *Score5*
  - RELIEF: *ParadigmId*, *EndsIn*, *LemmaSuffixProb*, *Score5*, *Score2*
- Some features consistently low-ranked (e.g. *POS*, *Score1*)
- **Multivariate feature subset selection**:
  - CFS: *StemSuffixProb*, *LemmaAttested*, *Score0*
  - CSS: . . . (13 features)

Evaluation – feature analysis

- Some features are **redundant** while others may be **irrelevant**
- Top-5 features with **univariate filter selection**:
  - IG: *StemSuffixProb*, *LemmaSuffixProb*, *Score6*, *Score5*, *Score7*
  - GR: *StemSuffixProb*, *LemmaSuffixProb*, *LemmaAttested*, *Score0*, *Score5*
  - RELIEF: *ParadigmId*, *EndsIn*, *LemmaSuffixProb*, *Score5*, *Score2*
- Some features consistently low-ranked (e.g. *POS*, *Score1*)
- **Multivariate feature subset selection**:
  - CFS: *StemSuffixProb*, *LemmaAttested*, *Score0*
  - CSS: . . . (13 features)

Evaluation – feature analysis

- Some features are **redundant** while others may be **irrelevant**
- Top-5 features with **univariate filter selection**:
  - IG: *StemSuffixProb*, *LemmaSuffixProb*, *Score6*, *Score5*, *Score7*
  - GR: *StemSuffixProb*, *LemmaSuffixProb*, *LemmaAttested*, *Score0*, *Score5*
  - RELIEF: *ParadigmId*, *EndsIn*, *LemmaSuffixProb*, *Score5*, *Score2*
- Some features consistently low-ranked (e.g. *POS*, *Score1*)
- **Multivariate feature subset selection**:
  - CFS: *StemSuffixProb*, *LemmaAttested*, *Score0*
  - CSS: . . . (13 features)

Evaluation – classification accuracy

take[lab];

Features (count)	Word-forms attested ≥ 1	≤ 100	≤ 10
All (22)	**91.97**	**91.94**	**90.65**
String-based (13)	87.01	87.69	87.98
Corpus-based (11)	87.78	86.59	82.04
IG (5)	81.14	79.05	76.46
GR (5)	59.76	80.90	77.29
RELIEF (5)	90.62	90.60	89.27
CFS (3)	81.69	79.51	78.67
CSS (13)	27.41	91.56	90.37
Baseline	50.00	56.51	69.92

Remarks

- How good can it guess the LPP? In reality, the set is imbalanced – must evaluate P and R on a per word basis
- Classifier confidence scores may be used to produce rankings (useful for semi-automatic lexicon enlargement)
- Evaluate w.r.t. size and diversity of the training set
- Consider additional evaluation scenarios: rule-based tagging, on-the-fly tagging, guessing paradigms of lemmas

Conclusion

take[lab];

- We framed paradigm guessing as a binary classification task over the output of a morphology grammar
- We defined 22 string- and corpus-based features
- Using all features gives the highest accuracy
- Using a subset of only five features gives almost as good results
- Decrease of accuracy on rare words is minimal
- FW: address the previously mentioned remarks

Thank you for your attention

take[lab];

Let's keep in touch. . .

www.takelab.hr

info@takelab.hr

Remember…

take[lab];