Experiments on Active Learning for Croatian Word Sense Disambiguation

Domagoj Alagić and Jan Šnajder
TakeLab UNIZG

Problem

- Many words are polysemous:
  - *The flight was delayed due to trouble with the* **plane**.
  - *Any line joining two points on a* **plane** *lies on that plane.*

Problem

- Many words are polysemous:
  - *The flight was delayed due to trouble with the* **plane**.
  - *Any line joining two points on a* **plane** *lies on that plane.*

Word Sense Disambiguation

Word sense disambiguation (WSD) is the task of computationally determining the meaning of a word in its context (Navigli, 2009).

WSD approaches

- **Knowledge-based WSD** vs. **supervised WSD**
- Supervised WSD systems give the best results
- However, they require large amounts of sense-annotated data as we need a separate classifier for each word
  $\Rightarrow$ extremely expensive and time-consuming
- Workaround: use both **labeled** and **unlabeled** data

Our work

- **Goal:** Cost-efficient WSD for Croatian
- **Objective:** Preliminary experiments using active learning (AL) for Croatian WSD
- **Methodology:**
  - Create a small manually-annotated lexical sample
  - Use simple supervised models with readily available features
  - Plug the models into an AL framework and evaluate their effectiveness (WSD accuracy) and efficiency (annotation effort reduction)
- **Contributions:**
  - First sense-annotated dataset for Croatian
  - Preliminary findings/recommendations on the use of various AL models on this dataset

Dataset

Corpus and sampling

- Croatian web corpus hrWaC (Ljubešić and Klubička, 2014) containing 1.9M tokens, lemmatized and MSD-tagged
- For the sense inventory, we have initially adopted the Croatian wordnet (CroWN), containing ~10k synsets
- We selected six polysemous words with 2 or 3 senses: **okvir**$_N$, **odlikovati**$_V$, **vatra**$_N$, **lak**$_A$, **brusiti**$_V$, **prljav**$_A$
- For each word, we sampled 500 sentences (contexts), yielding a total of 3000 word instances

Sense annotation

- 10 annotators
- 600 sentences (100 per word) per annotator
- Each word instance was double-annotated to obtain a more reliable annotation

Annotation guidelines

- Annotators were instructed to select a single word sense which they found the most appropriate for the given context, even in situations where multiple senses could be used
- For semantically opaque contexts (idioms, metaphors), we asked the annotators to choose the literate sense (e..g, "dirty laundry")
- In other cases (no adequate sense, erroneous instance), they were asked to select the "none of the above" (NOTA) option

Inter-annotator agreement

Word	κ	Word	κ
okvir$_N$	0.795	*odlikovati*$_V$	**0.978**
vatra$_N$	0.704	*lak*$_A$	**0.582**
brusiti$_V$	0.816	*prljav*$_A$	0.690

- Average Kappa coefficient of **0.761**
- Substantial variance in Kappa across the different words (indicative of sense overlaps, missing senses, etc.) $\Rightarrow$ FW

Gold standard sample

- Manually resolved all the disagreements
- In the majority of cases NOTA was among the responses
  $\Rightarrow$ CroWN incompleteness
- CroWN sense inventory modified to get a reasonable sense coverage on our lexical sample
- Total annotation effort: 36+6 hours

Dataset statistics

Word	Freq.	# Senses	Sense distr.	NOTA
okvir$_N$	141,862	2	381 / 115	4
vatra$_N$	45,943	3	244 / 106 / 141	9
brusiti$_V$	1,514	3	205 / 262 / 27	7
odlikovati$_V$	15,504	2	425 / 75	0
lak$_A$	15,424	3	277 / 87 / 113	23
prljav$_A$	14,245	2	228 / 187	85

Model

Active learning

- **Key idea:** allow the model to dynamically choose the instances from which it learns
- **Assumption:** by doing so the model can use fewer instances to achieve performance which is on par with the purely supervised models
- We use the **pool-based strategy** with **uncertainty sampling**
    - assumes that only those instances that carry the most information need to be labeled by an expensive human expert

Active learning loop

L : initial training set
U : pool of unlabeled instances
P : pool sample size
G : train growth size
f : classifier

while *stopping criteria not satisfied* **do**

    $f \leftarrow$ *train*(f, L);
    $R \leftarrow$ *randomSample*(U, P)
    predictions $\leftarrow$ *predict*(f, R)
    $R \leftarrow$ *sortByUncertainty*$(R,$ *predictions*$)$
    $S \leftarrow$ *selectTop*(R, G)
    $S \leftarrow$ *queryForLabels*(S)
    $L \leftarrow L \cup S$
    $U \leftarrow U \setminus S$

end

Active learning loop

L : initial training set
U : pool of unlabeled instances
P : pool sample size
G : train growth size
f : classifier
while *stopping criteria not satisfied* **do**
    $f \leftarrow$ *train*(f, L);
    $R \leftarrow$ *randomSample*(U, P)
    predictions $\leftarrow$ *predict*(f, R)
    $R \leftarrow$ *sortByUncertainty*$(R,$ *predictions*$)$
    $S \leftarrow$ *selectTop*(R, G)
    $S \leftarrow$ *queryForLabels*(S)
    $L \leftarrow L \cup S$
    $U \leftarrow U \setminus S$
end

Active learning loop

L : initial training set
U : pool of unlabeled instances
P : pool sample size
G : train growth size
f : classifier
while *stopping criteria not satisfied* **do**

$f \leftarrow \textit{train}(f, L)$;
$R \leftarrow \textit{randomSample}(U, P)$
$\textit{predictions} \leftarrow \textit{predict}(f, R)$
$R \leftarrow \textit{sortByUncertainty}(R, \textit{predictions})$
$S \leftarrow \textit{selectTop}(R, G)$
$S \leftarrow \textit{oracleLabel}(S)$
$L \leftarrow L \cup S$
$U \leftarrow U \setminus S$

end

Uncertainty sampling

1. **Least confident (LC):**

$$x^*_{\mathsf{LC}} = \underset{x}{\operatorname{argmax}} \left(1 - P_\theta(\hat{y}|x)\right)$$

2. **Minimum margin (MM):**

$$x^*_{\mathsf{MM}} = \underset{x}{\operatorname{argmin}} \left(P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x)\right)$$

3. **Maximum entropy (ME):**

$$x^*_{\mathsf{ME}} = \underset{x}{\operatorname{argmax}} \left(-\sum_i P_\theta(y_i|x) \log P_\theta(y_i|x)\right)$$

Classifier and features

Model:

- Core classifier: a linear Support Vector Machine (SVM) + fitted logistic curve at the output (Platt, 1999)
- Baseline: Most Frequent Sense (MFS) classifier

Features:

- Simple word-based context representations:
  1. Bag-of-words (BoW) – average dimension of $\sim$7000
  2. Skip-gram (SG) – 300 dimensions
- Feature vector computed by adding up the vectors of all content words from the context (sentence)

Results

Supervised baselines

- Random train-test split for each of the six words:
  400 instances for training and **100** for testing

Supervised baselines

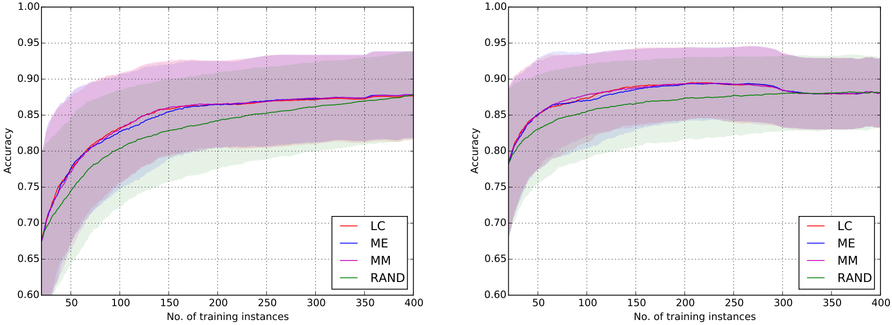
- Random train-test split for each of the six words: **400** instances for training and **100** for testing

Word	MFS	SVM-BoW	SVM-SG
okvir$_N$	0.53	**0.92**	0.89
vatra$_N$	0.49	**0.91**	0.88
brusiti$_V$	0.53	0.85	**0.86**
odlikovati$_V$	0.85	**0.97**	**0.97**
lak$_A$	0.55	0.80	**0.81**
prljav$_A$	0.46	0.82	**0.88**
Average:	0.57	**0.88**	**0.88**

Active learning experiments

- The same train-test split (400 train, 100 test)
- The initial training set L is a randomly chosen subset of the full training set
- Results averaged across 50 trials for each word
- Initial training set to 20, train growth size set to 1

Learning curves



(a) SVM-BoW

(b) SVM-SG

Active learning experiments

- All uncertainty sampling methods outperform RAND baseline ($\sim$2% points for 100 instances)
- All three uncertainty sampling methods perform comparably
- SVM-BoW: training on 100 instances gives $\sim$0.94% of the maximum accuracy (RAND requires twice that size)
- SVM-SG: training on 100 instances already gives the maximum accuracy

Parameter analysis

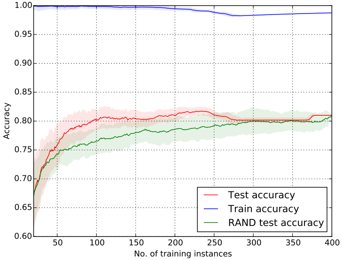
- A grid search over $L \in \{20, 50, 100\}$ and $G \in \{1, 5, 10\}$
- 300 runs per parameter pair (50 runs for each of the six words; $50 \times 6 = 300$)
- Area Under Learning Curve (ALC) – sum of accuracy scores across AL iterations normalized by the number of iterations

Parameter analysis

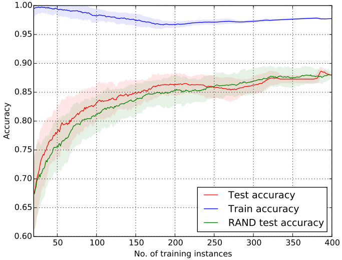
	G		
$\|L\|$	1	5	10
20	0.8794	0.8772	0.8760
50	0.8824	0.8819	0.8810
100	**0.8843**	0.8836	0.8833

- With larger L, more information is available to the learning algorithm up front
- With smaller G, model can make more confident predictions on yet unlabeled instances in each iteration

Per word analysis

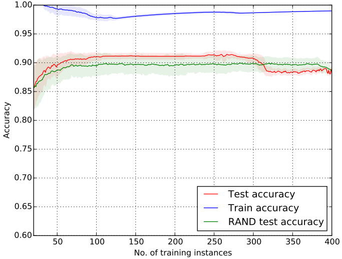


(a) lak_A (*easy*)

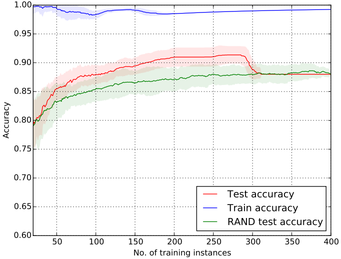


(b) $prljav_A$ (*dirty*)

Per word analysis

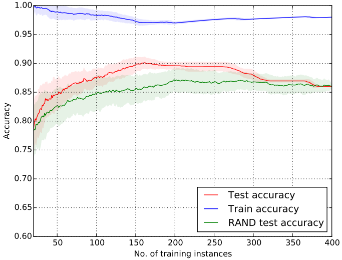


(a) *okvir*$_N$ (*frame*)

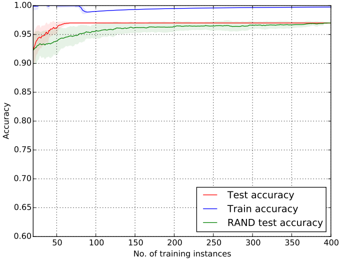


(b) *vatra*$_N$ (*fire*)

Per word analysis



(a) *brusiti*$_V$ (*to rasp*)



(b) *odlikovati*$_V$ (*to award*)

Per word analysis

- MM outperforms the RAND baseline for all six words
- AL gain is most prominent for *vatra*, *lak* and *brusiti*
  - full accuracy reachable with as few as **60** training instances
- For *prljav*, the learning curve does not saturate even after reaching 400 training instances
  $\Rightarrow$ too many NOTA labels?
- For *lak*, we observe the biggest train-test gap
  $\Rightarrow$ model overfits $\Rightarrow$ noisy dataset
  Low IAA? Non-informative contexts? Sense overlaps?

Per word analysis

- For some words the accuracy rises above that of a model trained on entire training set of 400 instances after which it drops
- Hypothesis: the model starts to overfit at some point (as we observe no drop in the training error)
- The subsequent drop in accuracy may be due to the sampling of a sequence of noisy instances from the training set
- Noise is likely not due to mislabeling (disagreements have been resolved), but rather due to non-informative contexts
- Should be further investigated

Conclusion

- On our 6-words dataset, uncertainty-based sampling AL gives 99% of accuracy of a fully supervised model at the cost of annotating only 100 instances
- On some words, AL model even outperforms a fully supervised model (when trained on a certain number of instances)

Conclusion

- On our 6-words dataset, uncertainty-based sampling AL gives 99% of accuracy of a fully supervised model at the cost of annotating only 100 instances
- On some words, AL model even outperforms a fully supervised model (when trained on a certain number of instances)

Future work:

- Lexical sample should be extended to enable more significant claims and recommendations
- Investigate issue of class imbalance
- Investigate stopping criteria
- Explore other uncertainty sampling methods
- Adapt to a noisy multi-annotator setup (crowdsourcing)

Thanks!

Dataset:
http://takelab.fer.hr/data/cro6wsd

TakeLab

http://takelab.fer.hr