Using Mined Coreference Chains as a Resource for a Semantic Task

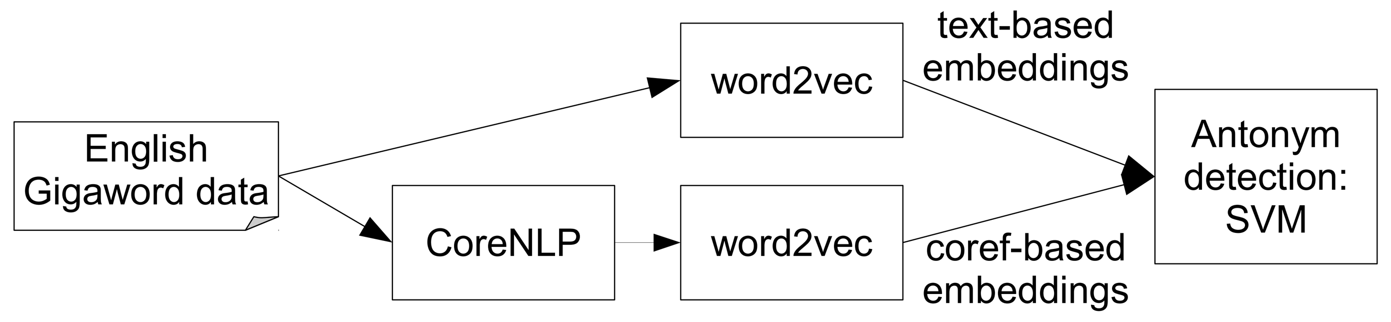
Heike Adel and Hinrich Schütze
Center for Information and Language Processing, University of Munich
heike.adel@cis.lmu.de

1. Introduction

- Motivation: use output of coreference resolution system as a resource for semantic tasks
- Coreference chains: **complementary** properties compared to other resources, such as cooccurrence statistics , e.g.: "cows" - "cattle" vs. "cows" - "milk"
- Coreference-based similarity can be used as an **additional feature** for any task that distributional similarity is useful for (e.g. finding alternative names for entities, knowledge base population)
- Task here: detecting antonyms

⇒ Antonyms: distributionally similar but semantically dissimilar words
⇒ Distributional models often cannot distinguish them from synonyms

2. Word embeddings



2.1 Word-based and coreference-based embeddings

- Calculation of word embeddings with word2vec (skip-gram model) [Mikolov et al., 2013]
- **text-based embeddings**: calculated on raw text data (English Gigaword, LDC2012T21, Agence France-Presse 2010) input to word2vec:

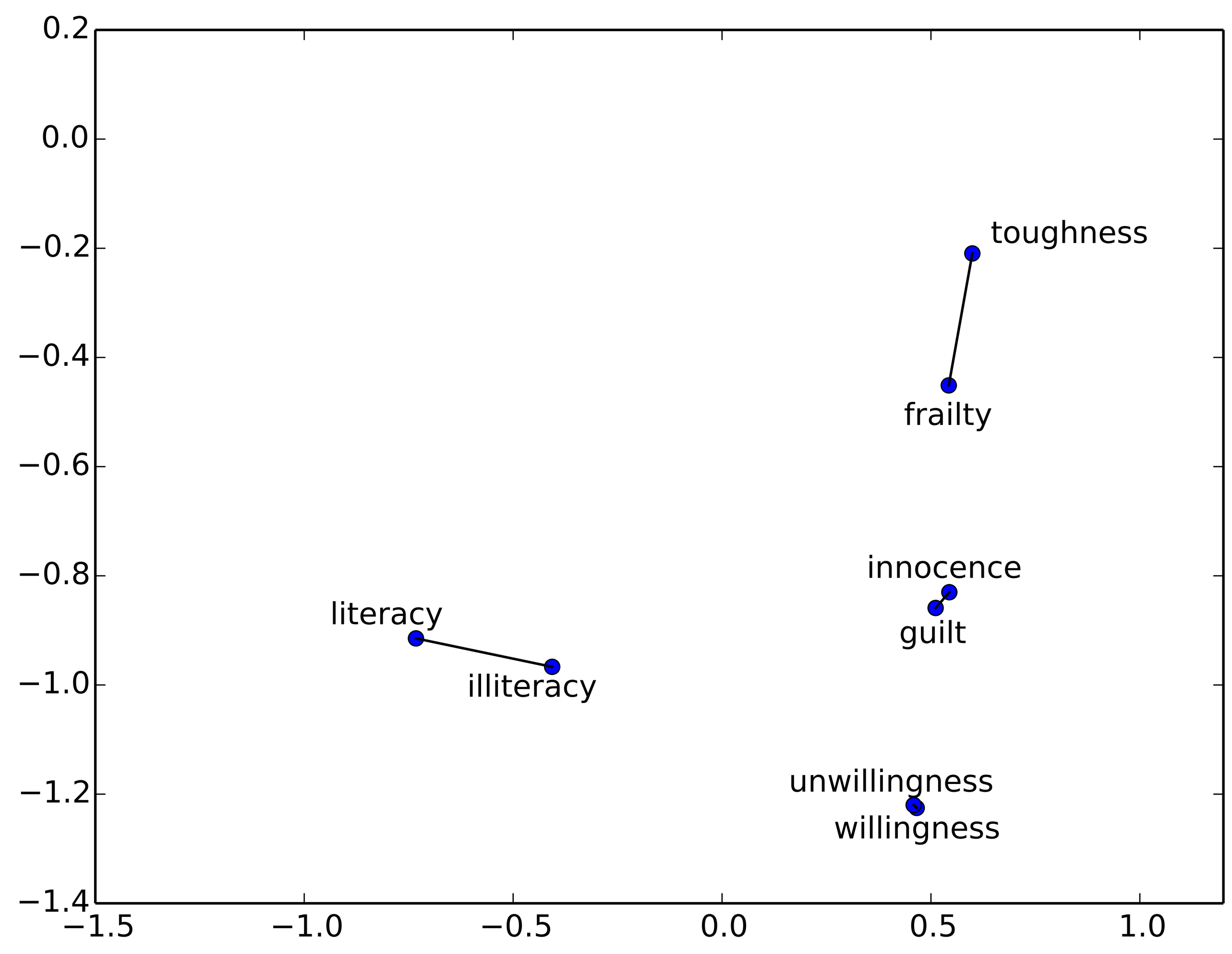
  Danish police late Friday shot and wounded a 27-year-old man trying to enter...
- **coreference-based embeddings**: calculated on automatically extracted coreference chains (one chain per line, coreference resolution with CoreNLP [Lee et al., 2011]) input to word2vec:

  Yusuf Mohammed ⇔ Mohammed ⇔ ruler of the Gulf state ⇔ ...
- Statistics:
  - 2.7M of 3.1M coreference chains are non-trivial
  - median (mean) length of chains: 3 (4.0) markables
  - median (mean) length of a markable: 1 (2.7) words
- Mined coreference chains: available at https://code.google.com/p/cistern

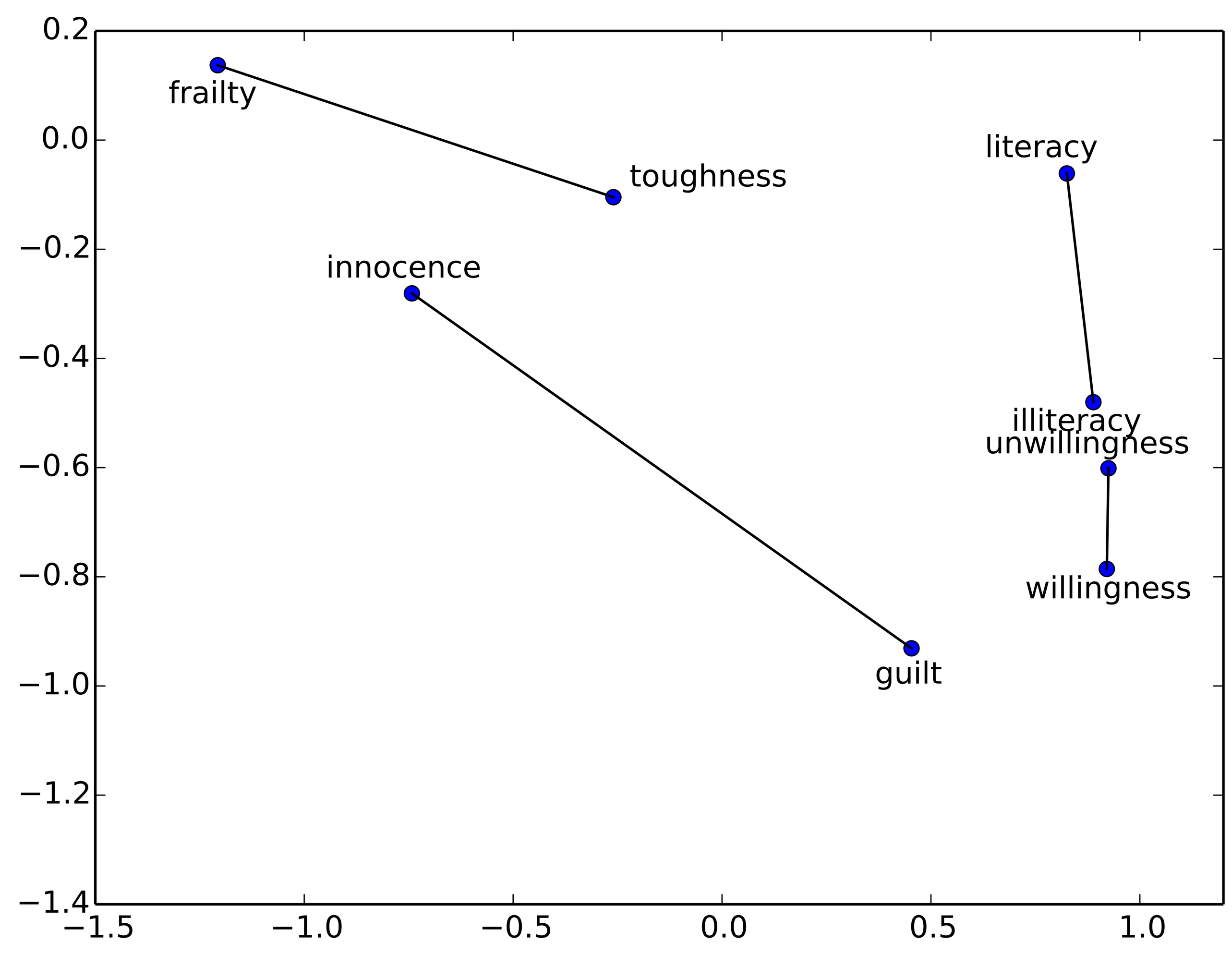
2.2 Qualitative analysis of word vectors

- Illustration after t-SNE [Van der Maaten and Hinton, 2008]

Text-based word embeddings (original cosine similarities: about 0.6)



Coreference-based word embeddings (original cosine similarities: about 0.05)



⇒ Coreference-based embeddings **enlarge the distance between antonyms**

- Five nearest neighbors-based on cosine similarity:

	text-based	coref.-based
his	my, their, her, your, our	he, him, himself, zechariah, ancestor
woman	man, girl, believer, pharisee, guy	girl, prostitute, lupita, betsy, lehia

⇒ Coreference-based neighbors: same gender
⇒ Substitution seems to change the meaning more for text-based neighbors than for coreference-based neighbors

2.3 Quantitative analysis of word vectors

- Split coreference resource into two parts (85% - 15%)
- First part: used for training embeddings
- Second part: used for computing cosine similarities for each possible word pair in the same coreference chain
- Results:

	minimum	maximum	median
text-based vectors	-0.350	0.998	0.156
coref.-based vectors	-0.318	0.999	0.161

⇒ Coreference-based vectors have **higher similarity within chains** than text-based vectors

3. Experiment: Antonym detection

3.1 Classification features

- Supervised classification with SVMs
- Features for SVM (to classify w and v as antonyms or non-antonyms):
  1. Cosine similarity of text-based embeddings of w and v
  2. Inverse rank of v in the nearest text-based neighbors of w
  3. Cosine similarity of coreference-based embeddings of w and v
  4. Inverse rank of v in the nearest coreference-based neighbors of w
  5. Difference of (1) and (3)
  6. Difference of (2) and (4)
- Feature subsets for experiments: text-based (1-2), coreference-based (3-4), all (1-6)

3.2 Data set

- Set of word pairs: target word w and antonym candidate v
- Possible target words: all word types of our vocabulary with at least one antonym in Merriam Webster [www.merriam-webster.com]
- Target words and their antonyms: available at https://code.google.com/p/cistern
- Positive training examples: target word and one of its antonyms which is also one of its 500 nearest text-based neighbors
- Negative training examples: same target word with a random word of its 500 nearest text-based neighbors

⇒ Idea: create a task that is **hard to solve** since all word pairs are **distributionally similar**

- In total: 2337 positive and 2337 negative examples
- Training set: 80%, validation set: 10%, evaluation set: 10%

3.3 Experimental results and discussion

	all word classification						noun classification					
	validation set			evaluation set			validation set			evaluation set		
feature set	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
text-based	**.83**	**.66**	**.74**	.74	.55	.63	**.91**	.61	.73	.74	.51	.60
coreference-based	.67	.42	.51	.65	.43	.52	.86	.47	.61	.77	.45	.57
text+coref	.79	.65	.72	**.75**	**.58**	**.66**	.88	**.70**	**.78**	**.79**	**.61**	**.69**

⇒ All word classification: coreference-based features: no improvements on validation set
⇒ All word classification: slightly better performance for combination of all features
⇒ Noun classification: using coreference-based features in addition to text-based features improves results
⇒ Mined coreference chains provide complementary information to cooccurrence statistics
  ⇒ **useful additional resource**
⇒ Reason why coreference-based embeddings alone perform worse than text-based embeddings alone:
  Different amount of training data:
  Coreference-chains: only a small subset of word-word relations encoded in raw text
⇒ More improvements for noun classification than for all word classification:
  Reason: e.g. adjectives with opposite meanings can cooccur in the same coreference chain
  For nouns: less likely since coreference chains contain markables referring to the same identical entity

4. Conclusion

- Coreference-based word embeddings capture a type of semantic similarity that is **complementary** to the one captured by text-based embeddings
- Coreference-based embeddings improve performance on antonym classification by .09 F_1

Acknowledgements

This work was supported by DFG (grant SCHU 2246/4-2).