Learning to Rank Lexical Substitutions

György Szarvas, Róbert busa-fekete, Eyke Hüllermeier
Presentation by Shushan Arakelyan, sa767@cam.ac.uk

Learning to rank lexical substitutions

Problem:
- Lexical substitution task

Given

Goal

Solution

Result

Lexical substitution task

Special form of contextual paraphrasing: replacing a single word

Lexical substitution subtasks:
- Generating possible substitutions
- Ranking candidate substitutions according to their contextual fitness

Learning to rank lexical substitutions

Problem

Given:
- Dataset of target words
- Sentential contexts
- Potential substitutions for the target words

Goal

Solution

Result

Datasets

Lexsub Dataset (McCarthy and Navigli, 2007)
- 201 target words (any part of speech)
- Contains 2002 sentences
- Lexical substitutions assigned to each (target word, sentence) pair by 5 native speakers

TWSI Dataset (Biemann, 2012)
- 1012 target nouns
- 24647 sentences
- Lexical substitutions for each target word in context from crowd sourced annotation

Potential Substitutions

WordNet *synsets*

- All synonyms
- *Similar to*
- *Entailment*
- *Also see*

Gold standard

Learning to rank lexical substitutions

Problem

Given

Goal:
- Train a machine learning model that accurately ranks the candidate substitutions based on their contextual fitness.

Solution

Result

Learning to rank lexical substitutions

Problem

Given

Goal

Solution:

- Several learning to rank methods, all using the same features.

Result

Delexicalized features

Local n-gram frequencies
- 1-5 gram frequencies extracted from web
- Syntagmatic coherence of the substitute in context

Corpus-based features
- Extracted from newspaper texts
- Non-local distributional features

Lexical resource features
- Extracted from WordNet

Shallow syntactic features
- Part of speech patterns

Classifiers (Part 1)

MaxEnt (Szarvas et al., 2013)

- Pointwise approach
- Formulates ranking as binary classification

ExpEns (Busa-Fekete et al., 2013)

- Pointwise approach with listwise meta-learning
- Listwise step uses AdaBoost

RankBoost (Freund et al., 2003)

- Pairwise boosting
- Optimizes the rank loss

Classifiers (Part 2)

RankSVM (Joachims, 2006)
- Paiwise approach, based on SVMs
- Formulates ranking as binary classification

LambdaMART (Wu et al., 2010)
- Listiwise approach
- Based on gradient boosted regression trees
- Gradient of parameters is calculated based on the evaluation metric

Learning to rank lexical substitutions

Problem

Given

Goal

Solution

Result:

- The performance on ranking task strongly depends on the way the task is formalized as a machine learning problem.

Experimental setup and evaluation

Experimental setup

- Cross validation on target word level

Evaluation

- Generalized Average Precision – the quality of the entire ranked list
- Precision at 1 - percentage of correct paraphrases at rank 1

Results (Part 1)

Database	LexSub		TWSI	
Candidates	WN	Gold	WN	Gold
	GAP			
MaxEnt	43.8	52.4	36.6	47.2
ExpEns	44.3	**53.5**	**37.8**	**49.7**
RankBoost	44.0	51.4	37.0	47.8
RankSVM	43.3	51.8	35.5	45.2
LambdaMART	**45.5**	**55.0**	**37.8**	**50.1**
	P@1			
MaxEnt	40.2	57.7	32.4	49.5
ExpEns	39.8	**58.5**	**33.8**	**53.2**
RankBoost	**40.7**	55.2	33.1	50.8
RankSVM	40.3	51.7	33.2	45.1
LambdaMART	**40.8**	**60.2**	**33.1**	**53.6**

Results (Part 2)

System	GAP
Erk and Padó (2010)	38.6
Dinu and Lapata (2010)	42.9
Thater et al. (2010)	46.0
Thater et al. (2011)	51.7
Szarvas et al. (2013)	52.4
ExpEns	53.5
LambdaMART	55.0

Thank you!

Questions?