Corpus Based Methods for Learning Models of Metaphor in Modern Greek

Pechlivanis Konstantinos
Konstantopoulos Stasinos

NCSR Demokritos - Institute of Informatics & Telecommunications (IIT)

25 November 2015

Contents

Introduction

Detecting metaphors:

- Detection, not interpretation
- Less-resourced languages
  - No access to semantic frames or networks
  - No access to full depth parser
- Assuming a purely statistical notion of semantics

Contribution:

- Minimization of the resources
- Broad thematic categories
- Require only the category of the article

Method

- Extracting knowledge from text classification models
  - Extract a measure of how characterstic of a domain is a term
    - Term Frequency - Inverse Document Frequency (tf-idf)
    - Method to estimate term weighting
  - Standard text classification techniques without any knowledge of metaphor
    - Maximum Likelihood Classifier (MLC)
    - Use only the term weighting
    - Highest weight $\Rightarrow$ classify term to equivalent domain

Corpus Collection

- One year's worth of crawling three newspapers
  - One offering CC content, two under license
  - Classify articles according to IPTC
  - Corpus will be made publicly available

IPTC code	Domain	Number	Percentage
01000000	Art, Culture and Entertainment	3178	20.2%
04000000	Economy,Business and Finance	3132	20.0%
06000000	Environment	693	4.4%
07000000	Health	771	4.9%
11000000	Politics	6618	42.2%
13000000	Science and Technology	210	1.3%
15000000	Sport	1100	7.0%
	All corpus	15702	

Table: Distribution of articles in topics.

Annotations(1/2)

- 10 articles manually annotated for testing
  - Two initial annotators
  - A third expert annotator create golder corpus
- annotation task
  - Read the whole text
  - Annotate domain of text
  - Annotate metaphor spans and the type of metaphors
- Metaphor types:
  - Multi-word metaphorical expression
  - Indirect metaphors
  - Direct (*is-a*) metaphors
  - Idiomatic metaphorical expressions

Annotations(2/2)

IPTC	All words	Content words	Metaphors
01000000	567	312	11
04000000	756	411	32
04000000	619	323	29
04000000	1158	650	34
07000000	321	169	13
11000000	760	414	51
11000000	961	518	52
11000000	715	404	15
11000000	985	558	50
11000000	987	546	53
All articles	7829	4305	340

Table: Annotated articles.

Implementation

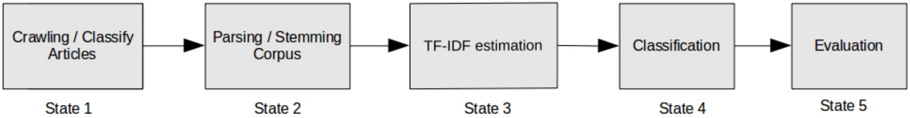

Figure: Processing Stages.

Corpus Preprocessing

- Tokenization
- Remove:
  - stopwords
  - tokens with alphanumeric characters
  - several symbols
  - stress marks
- Stemming - Implementation of a Greek Stemmer
  - available on-line
  - https://github.com/kpech21/Greek-Stemmer

TF-IDF

- tf (term frequency) ⇒ frequency of term in corpus
- idf (inverse document frequency) ⇒ number of documents that contains the term
- normalization factor ⇒ ensure that all values are between 0 and 1
- Treat all text of a domain as a single 'document'
- formula:

$$\begin{aligned} \text{tf-idf}(t,d) &= \text{tf}(t,d)\ \text{idf}(t,d) \\ &= \frac{\text{freq}(t,d)}{|T_d|} \log \frac{|D|}{|D_t|} \end{aligned}$$

term t, domain d, set of terms T_d, set of all domains D, set of domains where t appears D_t

TF-IDF

- tf (term frequency) ⇒ frequency of term in corpus
- idf (inverse document frequency) ⇒ number of documents that contains the term
- normalization factor ⇒ ensure that all values are between 0 and 1
- Treat all text of a domain as a single ‘document’
- formula:

$$\begin{aligned} \text{tf-idf}(t,d) &= \text{tf}(t,d)\ \text{idf}(t,d) \\ &= \frac{\text{freq}(t,d)}{|T_d|} \log \frac{|D|}{|D_t|} \end{aligned}$$

term t, domain d, set of terms T_d, set of all domains D, set of domains where t appears D_t

Classification

- Use of Maximum Likelihood Classifier (MLC):

$$\mathrm{MLC}(t, d_t) = \mathrm{argmax}_{d \in d_t} \text{tf-idf}(t, d)$$

  term t, set of domains where t appears d_t
- if $\mathrm{MLC}(t, d_t) == 0 \Leftrightarrow$
  - zero tf-idf value for all domains
  - unclassified term
- if $\mathrm{MLC}(t, d_t)\ != 0 \Leftrightarrow$
  - Term is classified in the domain where it appears the highest TF-IDF value

Evaluation: Precision, Recall, PoS

- Adapting Precision and Recall in our system
  - Precision:
    - is the percentage of positive decisions that were inside at least one span annotated as metaphor
  - Recall:
    - is the percentage of spans annotated as metaphors that include at least one positive decision
- Interaction with PoS(Part-of-Speech) features
  - Detection of Nouns, Verbs, Adjectives in articles
  - Use of Ellogon's part-of-speech tagger

Evaluation: terms with strong impact

- Words with the highest TF-IDF value
- From 24,463 classified words, use of 8,154 words
- X-Axis: top classified words
- Y-Axis: score of precision

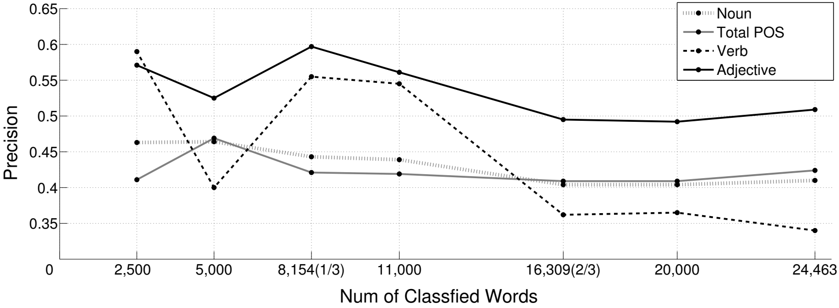

Figure: Precision results for the terms with the highest TF-IDF value.

Evaluation: Use of document frequency

- Vocabulary consist 28,305 unique words
- Classify all the words of vocabulary
  - 3,842 unclassified words
  - Classify the words with zero tf-idf according to the document frequency (df) and term frequency (tf)
  - tf is already estimated from tf-idf
- Document frequency of term t in document collection C:

$$\mathrm{df}(t, C) = \frac{\mathrm{freq}(t, C)}{|C|}$$

  - Threshold determined empirically
  - if df $<$ threshold :
    - ⋆ Classify term t according to the tf value
  - 2,037 additional classified words

Results

	All PoS	Noun	Adjective	Verb
Precision	0.443	0.421	0.597	0.555
Recall	0.285	0.150	0.209	0.066
$F_{\beta=1}$	0.347	0.221	0.300	0.119

Table: Evaluation results for the 1/3 of all the terms with the highest TF-IDF value.

	All PoS	Noun	Adjective	Verb
Precision	0.397	0.445	0.483	0.285
Recall	0.629	0.346	0.432	0.322
$F_{\beta=1}$	0.487	0.389	0.456	0.303

Table: Evaluation results for 26,500 classified words.

Comments

- First approach:
  - Scores the best Precision
  - Fewer classification results:
    - More accurate
    - Fewer detection (lower recall)
    - Same behavior for the PoS taggs
- Second approach:
  - Scores the best Recall
  - More classification results:
    - More detection
    - Less accurate (lower precision)
  - Scores the best F-score

Future Work

- N-gram
  - Use as terms: bigrams, trigrams and 4-grams
  - Check for each smaller gram too

Any questions?