Towards Determining Textual Characteristics of High and Low Impact Publications

Yue Chen, Kenneth Steimel, Everett Green, Nils Hjortnaes, Zuoyu Tian, Daniel Dakota, and Sandra Kübler

Outline

Introduction

- Predict future impact based on characteristics (i.e. text) of publication
- Impact is defined as citation count
  - We built a corpus for the purpose
  - 3 bins
- Which textual characteristics are found in successful papers?
- Initial results are mixed with tendency towards majority class
  - Class imbalance problem

Outline

Corpus Creation

- Articles from leading publications in the field of Computational Linguistics
  - Conference proceedings: ACL, NAACL, EACL and EMNLP
  - Journal papers: Computational Linguistics
- Obtaining the texts from the ACL Anthology Network Corpus
- Concentrating on two major topics
  - Parsing
  - Machine translation
- Collecting papers with two topics
  - Words “parse” or “parsing” in the title for parsing papers
  - Words “translate”, or “translation” in the title for MT papers
- Time period
  - Setting window size as 6 to 11 years ago
    i.e., we consider papers published between 2007 and 2012.

Distribution of Papers

Year	Total Papers	Parsing	Machine Translation
2007	187	83	104
2008	279	108	171
2009	270	135	135
2010	306	130	176
2011	191	67	124
2012	225	81	144

Outline

Experimental Setup

- Predicting ‘impact’ using an n-gram model
  - Using the 3 bins: 0 – 29, 30 – 119, 120 +
- Two types of experiments
  - Prediction using abstracts only
  - Prediction using full texts
  - Is the abstract as informative as the full paper?
- Experiment with removal of stopwords using a list of stopwords from nltk.corpus.stopwords

Abstract Extraction

- Used regular expression to extract text between “Abstract” and “Introduction”
- 70 papers did not match this pattern (Most of them end with “Motivation”)
  - These papers are mostly from the journal of Computational Linguistics
- These remaining papers had their abstracts extracted manually

Corpus splits

- Separated the two topics (Machine Translation and Parsing) during training and evaluation
- 10% development, 10% test, and 80% training

Outline

Feature Extraction

- The features extracted were simple word n-gram counts
  - Unigrams, bigrams, and trigrams

Feature Selection and Training

- Feature selection (filter methods)
  - χ^2-goodness of fit
  - Mutual Information
  - Variety of different feature selection thresholds
  - Also did prediction without feature selection
- A variety of different learning algorithms were used
  - Geometric methods
    - » Support Vector Machines
    - » KNN (TiMBL)
  - Ensemble methods
    - » Random Forests
    - » Gradient Boosting Trees
    - » Adaptive Boosting

Outline

Results using Abstracts

	Parsing			Machine Translation		
Classifier	# feats	Acc	F-score	# feats	Acc	F-score
Random Forest	All	60.61	45.74	All	67.82	56.87
	10 000	60.61	46.18	3 000	68.97	59.15
Gradient Boost	All	60.60	46.18	All	65.52	56.87
	5 000	60.60	48.91	10 000	67.82	58.89
Adaptive Boosting	All	60.60	45.74	All	66.67	58.89
	4 000	60.60	45.74	2 000	67.82	61.79
SVM	All	**62.12**	53.67	All	**68.97**	**65.66**
	10 000	**62.12**	**57.22**	10 000	**68.97**	**65.66**

Results using Full Texts

	Parsing			Machine Translation		
Classifier	# feats	Acc	F-score	# feats	Acc	F-score
ADABoost	1 000	**77.27**	**74.56**	2 000	**72.41**	**65.38**
SVM	50 000	68.18	60.35	50 000	71.26	64.43
Random Forest	2 000	71.21	65.56	1 000	71.26	64.50

Full Texts without Stopwords

	Parsing			Machine Translation		
Classifier	# feats	Acc	F-score	# feats	Acc	F-score
ADABoost	1 000	57.58	56.61	2 000	71.26	66.19
SVM	50 000	54.55	56.36	50 000	59.77	58.73
Random Forest	2 000	**66.67**	**60.00**	1 000	**74.71**	**72.44**

Overview of Results

- Majority prediction due to class imbalance
- Some feature sets + some classifiers help with abstracts
  - None of them successfully predicted the highest class
- Full text helps to predict the highest class
  - However removing stopwords helps with machine translation but not parsing

Outline

Feature Analysis

Parsing

- Time Window
  - CoNLL (2007)
  - Multilingual, dependency parsing, track
  - Systems from CoNLL
- Topic Modeling
  - May capture relevant time specific features
- Features may not be transferable to different time window

Feature Analysis

Machine Translation

- Not as clear
  - Stopwords
  - Generic words (system, evaluation, domain)
  - Non-distinguishable features
- Time Window
  - Joshua (2009) vs. Moses (2007)

Outline

Future Work

- More preprocessing
  - Feature engineering
    - Character n-grams
    - Dependency triples
  - Lemmatization
- Address class imbalance
  - Up-sampling
  - Split papers into more classes