Automatic annotation of text & Recognizing Textual Entailment

Basant Singh
Concordia University, Montreal, Canada

Introduction

Why NLP problems are hard ?

- Words can be ambiguous as its meaning depends on its context.
- Languages are changing everyday, so it is hard to define its problem by a given set of rules.

Creation of NLP based application

- Before advancement in Machine learning techniques, the task to create rules for language based application were done manually (by linguist).
- Increase in capacity and computation power of computer lead to the availability of unlimited text for the development of the application using machine learning techniques.
- After the advancement of Machine learning techniques, the hectic job of handwritten rules was given to the machine.
- Linguist were mostly required for labeling training data.

Challenge

- Using machine learning technique, we can create models but the vital task for NLP is to process text in order to create features.
- Ambiguity in language makes data processing and feature extraction a tough problem.

Goal

Process corpus to extract features and recognize textual entailment.

Feature Extraction Modules

Work on automatic annotation of text using NLTK
(Cornell movie reviews dataset) :

- **Sent_tokenize** :It break sentences based on punctuation marks.
  Issues : punctuation marks are often ambiguous.
  Solution: removal of unwanted punctuation marks.
  Issues : Unbalanced or unexpected parenthesis or bracket'.
  Solution: parse sentences to remove unbalanced parenthesis before invoking the module.

- **Word_tokenize**: split standard contractions, treat most punctuation characters as separate tokens, split off commas and single quotes, when followed by whitespace, separate periods that appear at the end of the line, for example
  Issues : it splits standard contraction like "don't" -> "do n't" but "n't" is not meaningful word.
  Solution: Replace such possible contraction with meaningful words.

- **pos_tag** : process a sequence of words, and attaches a part of speech tag to each word.
  Issue : Phrases are structurally ambiguous so POS of words that depends on words are ambiguous too. So POS tagging inherits this ambiguity in tagging words.
  Solution : We can create a model that can predict the possibility of pos tag for the word based on its context.

Named entity recognition using regular expressions

- **1st Approach** :After sentence splitting and tokenization only
  » Created regular expression based on pattern of words most common in Named entities.

- **2nd approach**: one, that runs after sentence splitting, tokenization, and POS tagging
  » POS gives additional information about words that is very critical for NER.

- **3rd approach** , that runs after sentence splitting, tokenization, POS tagging, and parsing.

  » Below table shows a comparison of numbers of token identified for the approaches.

dataset with number of tokens identified			
Data set	1st approach	2nd approach	3rd approach
Nokia_6610.txt	110	100	65
Nokia_6610-NLTK.txt	160	105	68
Movie_reviews (File id-1:300)	2274	50438	39078

Analysis of Named Entity recognition using regular expression

Below table shows Performance Evaluation of approaches for NER

1st Approach : Evaluation parameters						
dataset	tp	fp	fn	precision	recall	f-measure
Nokia_6610-NLTK.txt	7	4	2	.63	1	.38
Nokia_6610-NLTK.txt	14	1	1	.65	1	.93
2nd Approach : Evaluation parameters						
dataset	tp	fp	fn	precision	recall	f-measure
Nokia_6610-NLTK.txt	9	18	0	.33	1	.24
Nokia_6610-NLTK.txt	15	8	0	.65	1	.39
3rd Approach : Evaluation parameters						
dataset	tp	fp	fn	precision	recall	f-measure
Nokia_6610-NLTK.txt	9	12	0	.42	1	.29
Nokia_6610-NLTK.txt	13	4	2	.76	.86	.40

Implementation and design of RTE system

- Data reading:
  rte.pairs , nltk.word_tokenize , nltk.ne_chunk
- Data Processing :
  Cleaning , Annotation , Normalization
- RTE Feature Extractor:
  word_overlap , word_hyp_extra , ne_overlap, ne_hyp_extra, word_overlap_noun, word_overlap_verb, word_hyp_extra_noun, word_hyp_extra_verb, neg_txt, neg_hyp.
- Classification:
  » Experimented on 4 different classifiers to create a model and checked its performance on RTE test set and 2 addition new datasets.
  » Used Naive Bayes classifier to present result.
  » Found Naive Bayes algorithms performs better than the default maxEnt algorithm.
  » The performance of the classifier varied every time it was executed. For the classifier, could not find parameter for clasifier to get reproducible result .

Result and Analysis of RTE system

Experiments	RTE1				RTE2				RTE3				test1				test2			
	A	P	R	F score	A	P	R	F score	A	P	R	F score	A	P	R	F score	A	P	R	F score
Exp 1	56	57	56	56	56	58	57	56	63	63	63	63	70	70	70	69	40	45	40	32
Exp 2	55	56	55	55	56	56	56	54	61	62	62	62	60	60	60	60	53	63	53	51
Exp 3	52	52	52	52	56	57	56	56	61	61	61	61	60	60	60	60	46	57	47	42
Exp 4	55	56	55	54	57	57	57	56	63	63	63	63	70	70	70	69	46	57	47	42

Feature used :

- Exp1 : Word_overlap , word_hyp_extra, ne_overlap, ne_hyp_extra, neg_txt, neg_hyp
- Exp2 : Word_overlap_noun , word_overlap_verb, word_hyp_extra_noun, word_hyp_extra_verb
- Exp3 : Word_overlap , ne_overlap, neg_txt, neg_hyp, Word_overlap_noun word_overlap_verb
- Exp4 :Word_overlap , word_hyp_extra, ne_overlap, ne_hyp_extra, neg_txt, neg_hyp, ,Word_overlap_noun , word_overlap_verb, word_hyp_extra_noun, word_hyp_extra_verb

Conclusions

- Analyzed and built functionality for parsing and recognizing Named Entities.
- Functionality of the RTE System Developed.
- Among RTE datasets , RTE3, in expermient 4, shows performance measure of 63%. And for test1 it was 70%
- Scope of improvement
  » detection of negative sentiment in text or in hypothesis.
  » Issues in relation detection
  » Far away related words

References

[1] Rongzhou Shen, 2008 *Discovering Features Towards Recognising Textual Entailment* . University of Edinburgh.

[2] NLTK Classifier,
`https://www.nltk.org/_modules/nltk/classify/rte_classify.html`

[3] NLTK RTE,
`https://www.nltk.org/book/ch06.html`

[4] NLP Dictionary,
`http://www.cse.unsw.edu.au/~billw/nlpdict.html#case`

Note: It is incomplete list, full list is available in final report.