Uncertainty Learning Using SVMs and CRFs

Vinodkumar Prabhakaran
Computer Science Department
Columbia University, New York
vp2198@columbia.edu

Abstract

In this work, we explore the use of SVMs and CRFs in the problem of predicting certainty in sentences. We consider this as a task of tagging uncertainty cues in context, for which we used lexical, wordlist-based and deep-syntactic features. Results show that the syntactic context of the tokens in conjunction with the wordlist-based features turned out to be useful in predicting uncertainty cues.

1 Introduction

Extracting factual information from text is a critical NLP task which has important applications in Information Extraction, Textual Entailment etc. It is found that linguistic devices such as hedge phrases help to distinguish facts from uncertain information. Hedge phrases usually indicate that authors do not or cannot back up their opinions/statements with facts. As part of the CoNLL shared task 2010 (Farkas et al., 2010), we explored the applicability of different machine learning approaches and feature sets to learn to detect sentences containing uncertainty.

In Section 2, we present the task formally and describe the data used. Section 3 presents the system description and explains the features used in the task in detail. We investigated two different machine learning frameworks in this task and did experiments on various feature configurations. Section 4 presents those experiments and analyzes the results. Section 5 describes the system used for the shared task final submission and presents the results obtained in the evaluation. Section 6 concludes the paper and discusses a few future directions to extend this work.

2 Task Description and Data

We attempt only the Task 1 of the CoNLL shared task which was to identify sentences in texts which contain unreliable or uncertain information. In particular, the task is a binary classification problem, i.e. to distinguish factual versus uncertain sentences.

As training data, we use only the corpus of Wikipedia paragraphs with weasel cues manually annotated (Ganter and Strube, 2009). The annotation of weasel/hedge cues was carried out on the phrase level, and sentences containing at least one cue are considered as uncertain, while sentences with no cues are considered as factual. The corpus contained $11,110$ sentences out of which $2,484$ were tagged as uncertain. A sentence could have more than one cue phrases. There were 3143 cue phrases altogether.

3 System Description

3.1 Approach

We considered this task as a cue tagging task where in phrases suggesting uncertainty will be tagged in context. This is a 3-way classification problem at token level - B-cue, I-cue and O denoting beginning, inside and outside of a cue phrase. We applied a supervised learning framework for this task, for which We experimented with both SVMs and CRFs. For SVM, we used the Yamcha[1] system which is built on top of the tinySVM[2] package. Yamcha has been shown useful in similar tasks before. It was the best performing system in the CoNLL-2000 Shared task on chunking. In this task, Yamcha obtained the best performance for a quadratic kernel with a c value of 0.5. All results presented here use this setting. For CRF, we used the Mallet[3] software package. Experiments are done only with order-0 CRFs. CRFs proved to marginally improve the prediction accuracy while substantially improving the speed. For e.g, for a configuration of 10 features with context width of 2, Yamcha took around 5-6 hrs for 9-fold

[1] http://chasen.org/ taku/software/YamCha/

[2] http://chasen.org/ taku/software/TinySVM/

[3] http://mallet.cs.umass.edu/

Proceedings of the Fourteenth Conference on Computational Natural Language Learning: Shared Task, pages 132–137,
Uppsala, Sweden, 15-16 July 2010. ©2010 Association for Computational Linguistics

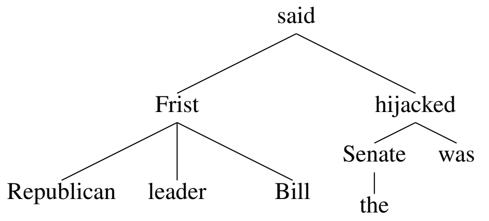
cross validation on the whole training set, where as Mallet took only around 30-40 minutes only.

3.2 Features

Our approach was to explore the use of deep syntactic features in this tagging task. Deep syntactic features had been proven useful in many similar tagging tasks before. We used the dependency parser MICA (Bangalore et al., 2009) based on Tree Adjoining Grammar (Joshi et al., 1975) to extract these deep syntactic features.

We classified the features into three classes - Lexical (L), Syntactic (S) and Wordlist-based (W). Lexical features are those which could be found at the token level without using any wordlists or dictionaries and can be extracted without any parsing with relatively high accuracy. For example, isNumeric, which denotes whether the word is a number or alphabetic, is a lexical feature. Under this definition, POS tag will be considered as a lexical feature.

Syntactic features of a token access its syntactic context in the dependency tree. For example, parentPOS, the POS tag of the parent word in the dependency parse tree, is a syntactic feature. The tree below shows the dependency parse tree output by MICA for the sentence *Republican leader Bill Frist said the Senate was hijacked.*



In this case, the feature haveReportingAncestor of the word *hijacked* is 'Y' because it is a verb with a parent verb *said*. Similarly, the feature haveDaughterAux would also be 'Y' because of daughter *was*, whereas whichAuxIsMyDaughter would get the value *was*.

Wordlist-based features utilized a list of words which occurred frequently as a cue word in the training corpus. We used two such lists – one which included adjectives like *many*, *most*, *some* etc. The other list contained adverbs like *probably*, *possibly* etc. The complete list of words in these wordlists are given in Table 1.

For finding the best performing feature set - context width configuration, we did an exhaustive search on the feature space, pruning away features which were proven not useful by results at stages.

The list of features we used in our experiments are summarized in Table 1 and Table 2. Table 1 contains features which were useful and are present in the results presented in section 4. Out of the syntactic features, parentPOS and isMyNNSparentGeneric turned out to be the most useful. It was noticed that in most cases in which a generic adjective (i.e., a quantifier such as *many, several, ...*) has a parent which is a plural noun, and this noun has only adjectival daughters, then it is part of a cue phrase. This distinction can be made clear by the below example.

- ⟨*ccue*⟩ *Many people* ⟨*/ccue*⟩ *enjoy having professionally made 'family portraits'*
- *Many departments, especially those in which students have research or teaching responsibilities ...*

In the first case, the noun *people* comes with the adjective *Many*, but is not qualified further. This makes it insufficiently defined and hence is tagged as a cue phrase. However in the second case, the clause which starts with *especially* is qualifying the noun *departments* further and hence the phrase is not tagged as a cue word despite the presence of *Many*. This scenario occurred often with other adjectives like *most*, *some* etc. This distinction was caught to a good extent by the combination of isMyNNSparentGeneric and isGenericAdj. Hence, the best performing configuration used features from both W and S categories.

The features which were found to be not useful is listed in Table 2. We used only two wordlist features, both of which were useful.

4 Experiments

To find the best configuration, we used 10% of the training data as the development set to tune parameters. Since even the development set was fairly large, we used 9-fold cross validation to evaluate each models. The development set was divided into 9 folds of which 8 folds were used to train a model which was tested on the 9th fold. All the reported results in this section are averaged over the 9 folds. We report $F_{\beta=1}$ (F)-measure as the harmonic mean between (P)recision and (R)ecall.

We categorized the experiments into three distinct classes as shown in Table 3. For each class, we did experiments with different feature sets and

No	Feature	Description
Lexical Features		
1	verbType	Modal/Aux/Reg (= 'nil' if the word is not a verb)
2	lemma	Lemma of the token
3	POS	Word's POS tag
4	whichModalAmI	If I am a modal, what am I? (= 'nil' if I am not a modal)
Word List Features		
1	isGenericAdj	Am I one of *some, many, certain, several*?
2	isUncertainAdv	Am I one of *generally, probably, usually, likely, typically, possibly, commonly, nearly, perhaps, often*?
3	levinClass	If I am a verb, which levin class do I belong to?
Syntactic Features		
1	parentPOS	What is my parent's POS tag?
2	leftSisPOS	What is my left sister's POS tag?
3	rightSisPOS	What is my right sister's POS tag?
4	whichModalIsMyDaughter	If I have a daughter which is a modal, what is it? (= 'nil' if I do not have a modal daughter)
5	Voice	Active/Passive (refer MICA documentation for details)
6	Mpos	MICA's mapping of POS tags (refer MICA documentation for details)
7	isMyNNSparentGeneric	If I am an adjective and if my parent is NNS and does not have a child other than adjectives
8	haveDaughterAux	Do I have a daughter which is an auxiliary.
9	whichAuxIsMyDaughter	If I have a daughter which is an auxiliary, what is it? (= 'nil' if I do not have an auxiliary daughter)

Table 1: Features used in the configurations listed in Table 4 and Table 6

Class	Description
L	Lexical features
LW	Lexical and Wordlist features
LS	Lexical and Syntactic features
LSW	Lexical, Syntactic and Wordlist features

Table 3: Experiment Sets

(linear) context widths. Here, context width denotes the window of tokens whose features are considered. For example, a context width of 2 means that the feature vector of any given token includes, in addition to its own features, those of 2 tokens before and after it as well as the prediction for 2 tokens before it. We varied the context widths from 1 to 5, and found that the best results were obtained for context width of 1 and 2.

4.1 Experimental Results

In this section, we present the results of experiments conducted on the development set as part of this task. The results for the system using Yamcha and Mallet are given in Table 4. CW stands for Context Width and P, R and F stands for Precision, Recall and F-measure, respectively. These results include the top performing 5 feature set - context width configurations using all three classes of features in both cases. It includes cue level prediction performance as well as sentence level prediction performance, where in a sentence is tagged as uncertain if it contains at least one cue phrase. In case of Mallet, it is observed that the best performing top 5 feature sets were all from the LSW category whereas in Yamcha, even configurations of LS category worked well.

We also present cue level results across feature categories for the Mallet experiments. Table 5 shows the best feature set - context width configuration for each class of experiments.

Class	Feature Set	CW
L	POS, verbType	2
LW	lemma, POS, modalMe, isGenericAdj, isUncertainAdj	2
LS	POS, parentPOS, modalDaughter, leftSisPOS, rightSisPOS, voice	2
LSW	POS, parentPOS, modalMe, isDaughterAux, leftSisPOS, mpos, isUncertainAdj, isGenericAdj, myNNSparentIsGeneric	1

Table 5: Best Feature sets - Across feature classes

Table 6 shows the cue level results of the best model for each class of experiments.

No	Feature	Description
		Lexical Features
1	Stem	Word stem (Using Porter Stemmer)
2	isNumeric	Word is Alphabet or Numeric?
		Syntactic Features
1	parentStem	Parent word stem (Using Porter Stemmer)
2	parentLemma	Parent word's Lemma
3	wordSupertag	Word's Super Tag (from Penn Treebank)
4	parentSupertag	Parent word's super tag (from Penn Treebank)
5	isRoot	Is the word the root of the MICA Parse tree?
6	pred	Is the word a predicate? (pred in MICA features)
7	drole	Deep role (drole in MICA features)
8	haveDaughterTo	Do I have a daughter 'to'?
9	haveDaughterPerfect	Do I have a daughter which is one of *has, have, had*?
10	haveDaughterShould	Do I have a daughter *should*?
11	haveDaughterWh	Do I have a daughter who is one of *where, when, while, who, why*?

Table 2: Features which turned out to be not useful

Class	Cue P	Cue R	Cue F
L	54.89	21.99	30.07
LW	51.14	20.70	28.81
LS	52.08	25.71	33.23
LSW	51.13	29.38	36.71

Table 6: Cue level Results - Across feature classes

4.2 Analysis

It is observed that the best results were observed on LSW category. The main constituent of this category was the combination of isMyNNSparentGeneric and isGenericAdj. Also, it was found that W features used without S features decreased the prediction performance. Out of the syntactic features, parentPOS, leftSisPOS and rightSisPOS proved to be the most useful in addition to isMyNNSparentGeneric.

Also, the highest cue level precision of 54.89% was obtained for L class, whereas it was lowered to 51.13% by the addition of S and W features. However, the performance improvement is due to the improved recall, which is as per the expectation that syntactic features would help identify new patterns, which lexical features alone cannot. It is also worth noting that addition of W features decreased the precision by 3.75 percentage points whereas addition of S features decreased the precision by 2.81 percentage points. Addition of S features improved the recall by 3.72 percentage points where as addition of both S and W features improved it by 7.39 percentage points. However, addition of W features alone decreased the recall by 1.29 percentage points. This suggests that the words in the wordlists were useful only when presented with the syntactic context in which they occurred.

Mallet proved to consistently over perform Yamcha in this task in terms of prediction performance as well as speed. For e.g, for a configuration of 10 features with context width of 2, Yamcha took around 5-6 hrs to perform the 9-fold cross validation on the entire training dataset, whereas Mallet took only around 30-40 minutes.

5 System used for Evaluation

In this section, we explain in detail the system which was used for the results submitted in the shared task evaluation.

For predicting the cue phrases on evaluation dataset for the shared task, we trained a model using the best performing configuration (feature set and machinery) from the experiments described in Section 4. The best configuration used the feature set <POS, parentPOS, modalMe, isDaughterAux, leftSisPOS, mpos, isUncertainAdj, isGenericAdj, myNNSparentIsGeneric> with a context width of 1 and it was trained using Mallet's CRF. The cross validation results of this configuration is reported in Table 4 (First feature set in the Mallet section). This model was trained on the entire Wikipedia training set provided for Task 1. We used this model to tag the evaluation dataset with uncertainty cues and any sentence where a cue phrase was tagged was classified as an uncertain sentence.

Feature Set	CW	Cue			Sent		
		P	R	F	P	R	F
Yamcha - Top 5 Configurations							
POS, parentPOS, modalDaughter, leftSisPOS, rightSisPOS, levinClass, myNNSparentIsGeneric	2	51.59	26.96	34.10	65.27	38.33	48.30
POS, parentPOS, amIuncertain	1	43.13	29.41	33.79	55.37	41.77	47.62
POS, parentPOS, modalDaughter, leftSisPOS, rightSisPOS, voice	2	52.08	25.71	33.23	66.52	37.10	47.63
POS, parentPOS, modalDaughter, leftSisPOS	2	54.25	25.16	33.20	69.38	35.63	47.08
POS, parentPOS, modalDaughter, leftSisPOS, rightSisPOS, mpos	2	51.82	25.56	33.01	65.62	36.12	46.59
Mallet - Top 5 Configurations							
POS, parentPOS, modalMe, isDaughterAux, leftSisPOS, mpos, isUncertainAdj, isGenericAdj, myNNSparentIsGeneric	1	51.13	29.38	36.71	66.29	42.71	51.95
POS, parentPOS, modalMe, isDaughterAux, leftSisPOS, mpos, voice, isUncertainAdj, isGenericAdj, myNNSparentIsGeneric	1	49.81	29.07	36.04	65.64	42.24	51.40
POS, parentPOS, modalMe, isUncertainAdj, isGenericAdj, myNNSparentIsGeneric	2	52.57	28.96	35.55	65.18	39.56	49.24
POS, parentPOS, modalMe, auxDaughter, leftSisPOS, mpos, voice, isUncertainAdj, isGenericAdj, myNNSparentIsGeneric	1	48.22	28.67	35.40	65.25	42.80	51.69
POS, parentPOS, modalMe, leftSisPOS, mpos, voice, isUncertainAdj, isGenericAdj, myNNSparentIsGeneric	1	52.26	28.12	35.34	65.99	40.05	49.85

Table 4: Overall Results

5.1 Evaluation Results

This section presents the results obtained on the shared task evaluation in detail. The sentence level results are given in Table 7. Our system obtained a high precision of 87.95% with a low recall of 28.42% and F-measure of 42.96% on the task. This was the 3rd best precision reported for the Wikipedia task 1.

System	Precision	Recall	F-Measure
Best System	72.04	51.66	60.17
...	...	...	...
This System	**87.95**	**28.42**	**42.96**
Last System	94.23	6.58	12.30

Table 7: Evaluation - Cue Level Results

Table 8 presents the cue level results for the task. Our system had a cue level prediction precision of 67.14% with a low recall of 16.70% and F-measure of 26.75%, which is the 3rd best F-measure result among the published cue level results[4].

We ran the best model trained on Wikipedia corpus on the biomedical evaluation dataset. As expected, the results were much lower. It obtained a precision of 67.54% with a low recall of 19.49% and F-measure of 30.26%.

System	Precision	Recall	F-Measure
X	63.01	25.94	36.55
X	76.06	21.64	33.69
This System	**67.14**	**16.70**	**26.75**
X	28.95	14.70	19.50
X	24.57	7.35	11.32

Table 8: Evaluation - Cue Level Results

6 Conclusion and Future Work

A simple bag of words approach at the sentence level could have given similar or even better performance for the sentence level prediction task. However, identifying cues in context is important to extend this task to application where we need to make semantic inferences or even identifying the scope of uncertainty (which was the task 2 of the shared task). Hence, we infer that this or a similar cue tagging approach with a more sophisticated feature set and machinery should be explored further.

Our experiments show that the addition of syntactic features helps in improving recall. However, the advantage given by syntactic features were surprisingly marginal. In detailed error analysis, it was found that the syntactic patterns that proved helpful for this task were fairly local. So, probably exploring shallow syntactic features instead of deep syntactic features might be helpful for this task. Also, we assume that using more sophis-

[4] In the submitted result, cues were tagged in IOB format. Hence, cue level statistics were not computed and published in the CoNLL website.

ticated lexical features or custom made lexicons could also improve performance.

Acknowledgements

This work was supported by grants from the Human Language Technology Center of Excellence. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the sponsor.

I would also like to extend my heartfelt gratitude to Prof. Kathy McKeown and Yves Petinot for their vital encouragement and support throughout this project. I would also like to thank my advisors Dr. Mona Diab and Dr. Owen Rambow for their valuable suggestions and support.

References

Srinivas Bangalore, Pierre Boullier, Alexis Nasr, Owen Rambow, and Benoît Sagot. 2009. MICA: A probabilistic dependency parser based on tree insertion grammars. In *NAACL HLT 2009 (Short Papers)*.

Richárd Farkas, Veronika Vincze, György Móra, János Csirik, and György Szarvas. 2010. The CoNLL-2010 Shared Task: Learning to Detect Hedges and their Scope in Natural Language Text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning (CoNLL-2010): Shared Task*, pages 1–12, Uppsala, Sweden, July. Association for Computational Linguistics.

Viola Ganter and Michael Strube. 2009. Finding Hedges by Chasing Weasels: Hedge Detection Using Wikipedia Tags and Shallow Linguistic Features. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 173–176, Suntec, Singapore, August. Association for Computational Linguistics.

Aravind K. Joshi, Leon Levy, and M Takahashi. 1975. Tree adjunct grammars. *Journal of the Computer and System Sciences*, 10:136–163.