Word Class Lattices for Definition Extraction

B Arjun and Shubhangi Ghosh

Indian Institute of Technology Madras

Abstract. Definition extraction is the task of automatically identifying definitional sentences within texts. In this project, we have implemented and analyzed the Word Class Lattice approach to definition extraction, as described in [2]. Based on our analysis of these models we have proposed and implemented modifications to the WCL model which make use of Syntactic Knowledge and Context Free Grammar Rules, which have outperformed the existing WCL models.

1 Introduction

Textual definitions are an essential source of information when looking up meanings of terms. It is, however, very difficult to manually obtain definitions and update existing dictionaries as it involves experts in the particular field to verify it. It is observed that new terms will usually be present in text along with sentences which contain the definition of the term. This motivates a requirement to automate the task of extracting definitions from this data using Machine Learning (ML) and Natural Language Processing techniques.

This project explores the method of Learning Word Class Lattices for Definition and Hypernym Extraction. We also propose some variants to the algorithm to improve upon some of its drawbacks, after analyzing the sentences which it fails to classify correctly.

2 Related Approaches

The majority of the approaches prior to Word Class Lattices use symbolic methods that depend on lexico-syntactic patterns or features, which are manually crafted or semi-automatically learned. These suffer from low recall and low precision. In this project we use a generalized form of word lattices, called Word-Class Lattices (WCLs), as an alternative to lexico-syntactic pattern learning. A lattice is a directed acyclic graph (DAG), a subclass of non-deterministic finite state automata (NFA). The lattice structure has the purpose of preserving the salient differences among distinct sequences, while eliminating redundant information.

3 Word Class Lattices

3.1 Structure of Definitions

Given a definition we assume that it contains the following fields:

- The DEFINIENDUM field (DF): this part of the definition includes the *definiendum* (that is, the word being defined) and its modifiers (e.g., In computer science, a closure)
- The DEFINITOR field (VF): it includes the verb phrase used to introduce the definition (e.g., is)
- The DEFINIENS field (GF): it includes the genus phrase (usually including the hypernym, e.g., a first-class function)
- The REST field (RF): it includes additional clauses that further specify the *differentia* of the definiendum with respect to its genus (e.g., with free variables that are bound in the lexical environment)

3.2 Sentence Generalisation

Word Classes We use the set of frequent words F, from nltk.corpus.stopwords to generalize words to word classes. We define a word class as either a word itself(if it belongs to the list of frequent words, F) or its part of speech. The word being defined is replaced by TARGET, which also comes in the list of frequent words, F.

$$w_i = \begin{cases} w_i, & \text{if } w_i \in F \\ POS(w_i), & \text{otherwise} \end{cases}$$

Generalised Sentences Generalised sentences are constructed by replacing each word in a sentence by its word class, i.e. Part of Speech.

3.3 Construction

Star Patterns Every definitional sentence in the training set is assigned a star pattern.

1. The words in the definition that don't belong to F are replaced by '*'.
2. The words in the definition that belong to the set F, are retained as is.
3. The REST FIELD (RF) is ignored.

Sentence Clustering A clustering of training sentences, $C = C_1, C_2 ... C_n$ is formed, where each cluster C_i contains sentences belonging to a particular star pattern.

Word-Class Lattice Construction

1. Consider a cluster $C_i = \{s_1, s_2, ...s_{|C_i|}\}$. Consider the first sentence in C_i, $s_1 = w_1^1, w_2^1, ...w_{|s_1|}^1$, and construct the corresponding generalised sentence, $s_1^{'} = w_1^1, w_2^1, ..., w_{|s_1|}^1$.

2. Construct a directed graph $G = (V, E)$ such that $V = \{w_1^1, w_2^1, ...w_{|s_1|}^1\}$ and $E = (w_1^1 w_2^1), (w_2^1, w_3^1)...(w_{|s_1|-1}^1, w_{|s_1|}^1)$
3. Now, we use dynamic programming to add consequent sentences to the graph. For each following sentence, we compute alignment score with all sentences in the cluster preceding it.The alignment score between two sentences s_j and s_k is computed as:
   (a) $M_{a,b} = max(M_{a-1,b-1} + S_{a,b}, M_{a-1,b}, M_{a,b-1})$, where $a \in 1, 2...s_k$, $b \in 1, 2...s_j$.
   (b) $M_{0,0}, M_{0,b} and M_{a,0}$ are initially set to 0 for all a and b.
   (c) The matching score $S_{a,b}$ is calculated on the generalized sentences $s_k^{'}$ and $s_j^{'}$ as follows:

$$S_{a,b} = \begin{cases} 1, & \text{if } w_a^k = w_b^j \\ 0, & \text{otherwise} \end{cases}$$

   (d) Finally, the alignment score between s_k and s_j is given by $M_{|s_k||s_j|}$.
4. The sentence $s_k(k < j)$ with best alignment to s_j is chosen to add s_j to the graph. The set of vertices not already present in the lattice ar added and the edges present between consecutive generalised tokens in s_j are added to the graph.
5. Furthermore, in the final lattice, nodes associated with the hypernym words in the learning sentences are marked as hypernyms in order to be able to determine the hypernym of a test sentence at classification time.

3.4 WCL-1

In this model the lattices are learnt from training sentences in their entirety, i.e. each lattice is used to match a complete sentence, and not different sections of it. The disadvantage of this approach is that there is very high variability in the syntactic structure of definitional sentences, so it could lead to lower recall.

Proposed Variants

Pattern Augmentation In order to improve the recall of WCL-1, based on grammatic rules, extra generalized sentences were created for each sentence in the training set, which was then added to the WCL to improve generalization.

For example consider a sentence of the form 'TARGET is a type of JJ NN', where JJ represents an adjective and NN represents a noun. If this sentence is seen in the training set, the corresponding generalized sentence would be incorporated into the Word Class Lattice. However a test sentence of the form 'TARGET is a type of NN' would not get matched, even though they share alot of similarity in terms of the grammar. Essentially, both JJ NN, and NN form noun phrases and thus their behaviour should be similar. However the WCL model does not capture this, resulting in many False Negatives.

The additional patterns added to our augmentation list included the following:

- Replacing occurrences of NN NN and NN NN NN with NN and vice versa.
- Replacing occurrences of JJ NN with NN and vice versa.
- Replace occurrences of 'the TARGET', 'a TARGET' and 'an TARGET' with TARGET, and vice versa.
- Replacing occurrences of 'the NN', 'a NN' and 'an NN' with NN, and vice versa.
- Replacing occurrences of JJ JJ with JJ and vice versa. (For example, a wonderful beautiful place, a wonderful place)
- Replacing NNS with NN and vice versa, where NNS is the tag for plural nouns.
- Replacing occurrences of 'the' with 'a' and vice versa.
- Replacing occurrences of NN with NP and vice versa, where NP is the tag for Proper Nouns.
- Replacing occurrences of NP NP with NP and vice versa.
- Replacing occurrences of RB VV with VV, where RB is the tag for Adverbs. (For example: TARGET refers to efficiently studying for 5 hours, TARGET refers to studying for 5 hours.)

To improve generalization further, augmentation was also applied at the test sentence side, i.e. for any test sentence occurrences of NN NN were replaced with NN, JJ JJ with JJ, JJ NN with NN, and RB VV with VV.

Match Based Scoring In order to improve the recall of WCL-1, we chose to use a softer classification criterion as compared to strictly checking the patterns. For example, sentences like 'TARGET refers to NN NN NN' would not get classified as definitional even if the training set contained a sentence like 'TARGET refers to JJ NN NN'. This motivated the need to keep a track of the number of matching tokens with the WCL.

For each sentence in the test set, we first identify the star pattern and choose the corresponding WCL for that sentence cluster. For this WCL, we compute the match score of the test sentence with each of the stored generalized sentences in the WCL, and classify the sentence as definitional if the maximum score obtained crosses a threshold which is proportional to the length of the sentence.

Coverage-Support based Scoring Based on the ideas of WCL-3 and the Match based idea mentioned in the previous paragraph, we chose to use a scoring function based on the Coverage and Log Support product used for WCL-3.

$$score(s, WCL) = coverage \times log(support)$$

where s is the candidate sentence, WCL is the lattice being checked, coverage is the fraction of words of the input sentence covered by the lattice, and support is the sum of the number of sentences in the star patterns corresponding to the lattice.
The sentence was classified as definitional if this score was larger than a threshold, which was a hyperparameter. This allows for more generalization, but may also result in non-definitional sentences being classified as definitional.

3.5 WCL-3

Separate lattices are learnt for defineidum(DF), definitor(VF) and definiens(GF) fields. This is done as definitional patterns in the entire sentence may exhibit higher variability as compared to definitional patterns in the sub-fields. Thus, WCL-3 improves generalization power, thereby giving better recall in most cases.

Proposed Variants

Pattern Augmentation Similar to the case of WCL-1, based on grammatic rules, extra generalized sentences were created for each sentence in the training set, which was then added to the WCL to improve generalization.

Field-Specific Coverage-Support Score Instead of computing a single coverage score for all 3 fields' Lattices together, in this model, a coverage-support score was computed for each field separately, and the maximum score for each was used. This greedy approach, is more efficient in terms of computation time, as it doesn't involve iterating through every WCL multiple times.

4 Results

4.1 Dataset Used and Training/Test Details

The dataset used for training and evaluating the model was a corpus of 4,619 Wikipedia sentences, that contains 1,908 definitional and 2,711 non-definitional sentences. The former were obtained from a random selection of the first sentences of Wikipedia articles. The defined terms belong to different Wikipedia domain categories, so as to capture a representative and cross-domain sample of lexical and syntactic patterns for definitions. These sentences were manually annotated with DEFINIENDUM, DEFINITOR, DEFINIENS and REST fields by an expert annotator, who also marked the hypernyms. The associated set of negative examples (syntactically plausible false definitions) was obtained by extracting from the same Wikipedia articles sentences in which the page title occurs.

However, the paper did not mention the exact train-test split used in their evaluation of the model, and thus our scores don't perfectly tally with theirs. In case of WCL-3, they haven't mentioned the threshold used for the coverage score because of which our score doesn't match theirs completely. We used 50% of the definitional sentences for training, and the rest along with the non-definitional sentences for testing. The paper reported the (P,R,F1) Scores for WCL-1 as (0.9988,0.4209,0.5922) and WCL-3 as (0.9881, 0.6074,0.7523)

4.2 Evaluation Measures

- Precision: The number of definitional sentences correctly retrieved by the system over the number of sentences marked by the system as definitional.
- Recall: The number of definitional sentences correctly retrieved by the system over the number of definitional sentences in the dataset.
- F_1-measure: The harmonic mean of precision(P) and recall(R), given by $\frac{2PR}{P+R}$.

5 Analysis

Table 1. Performance of WCL and its variants for Definition Extraction on Wikipedia Dataset

Model Name	Precision	Recall	F-Score
WCL-1	0.9965	0.3034	0.4652
WCL-1 with Pattern Augmentation	0.9977	0.4637	0.6331
WCL-1 with Pattern Augmentation and Match based Score	0.9980	0.5459	0.7058
WCL-1 with Coverage-Support Score	0.5977	0.8002	0.6843
WCL-1 with Pattern Augmentation and Coverage-Support Score	0.9489	0.6599	0.7757
WCL-3	0.7426	0.6720	0.7055
WCL-3 with Pattern Augmentation	0.5362	0.9006	0.6722
WCL-3 with Field-Specific Coverage-Support	0.6612	0.7303	0.7015

The first thing to note is that our WCL-1 F-Score was off by about 13% as compared to the paper, and WCL-3 F-Score was off by 5%.

In Table 1, we observe that WCL-1 with Pattern Augmentation and Coverage-Support Score performs the best, surpassing the WCL-1 F-score reported in the paper by 18% and the WCL-3 F-Score by 2%. As compared to our WCL-1 F-Score, this model had a 31% increase in score.

We observe that WCL-1 has very high precision but poor recall. This is because of the small number of true positives, as it predicts very few definitional sentences. By applying the pattern augmentation model, the recall improved by 16%, and by adding the match function it increased totally by 24%. This is because the model generalizes better after adding these, and it classifies more sentences as definitional. We expected the precision to drop, however this did not occur as most of the non-definitional sentences follow different star patterns itself as compared to the positive sentences. Based on the WCL-3 Coverage-Support score along with Pattern Augmentation the recall improved by 35%, though the precision dropped by 5%. This improvement occurs as the Coverage-Support score is more natural for this case, and it allows more sentences to be classified definitional as the sentence does not have to completely match the star pattern, instead we look at the fraction covered. This also explains the drop in precision as some non-definitional sentences get misclassified.

In case of WCL-3, the default model performs the best as compared to the one with pattern augmentation. This is because the WCL-3 precision is already lower as it is designed as a generalization of WCL-1. The second reason for pattern augmentation worsening the result is because the individual variation in each of the phrases is less, thus by using augmentation in each phrase too many spurious sentences are added to each WCL.

The Field-Specific Coverage WCL-3 performs worse than the default one, as it is a greedy approximation, and the maximum score for each field may not be the maximum score over all the fields. However the Field Specific variant is much faster as it does not have to iterate multiple times over all the WCLs.

5.1 Variation of WCL-3 Performance with Threshold

As seen in Figure 1, we varied the coverage threshold to see the change in performance of WCL-3. For low thresholds the precision is very low, as low coverage matches are also classified as definitional. However with increase in threshold, the precision increases but recall drops. Thus we found the optima with respect to F-Score at Threshold=3.

6 Conclusion

By analyzing the WCL models, we observed that we can improve the generalization by adding more syntactic knowledge to the model, such as pattern augmentation. This could be further improved by using CFG Non terminals like Noun Phrase or Verb Phrase, etc instead of the final POS tags. This would be a better generalization, so that text of the form JJ NN, JJ JJ NN, and so on would match, as all are noun phrases.

Another factor to be considered is that definition extraction involves non-conscious knowledge to a certain degree. In the paper [1], the authors used LSTMs with generalised sentences(with POS tags) to obtain 0.912 F-Score, surpassing all the previous state of the art models. By combining grammatical knowledge and finding patterns in this using ML/DL ideas, it is possible to extract definitions with a higher degree of accuracy.

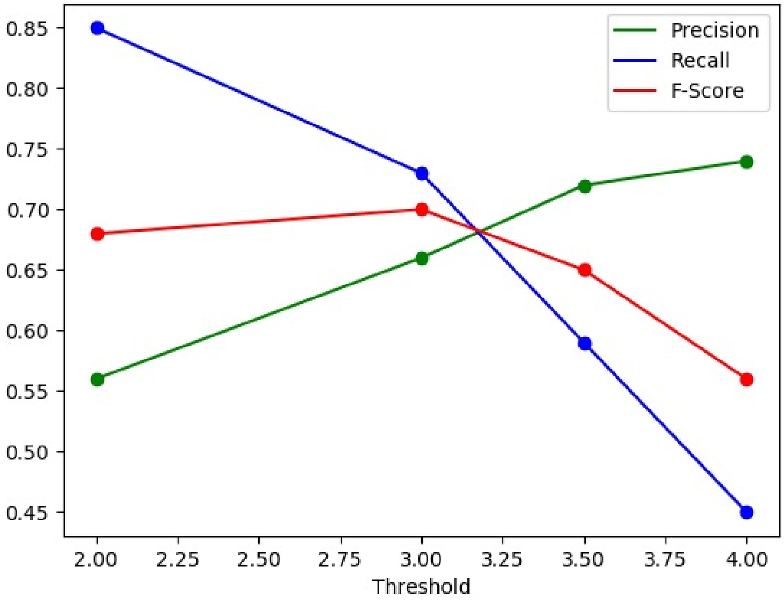

Fig. 1. Variation of Performance of WCL-3 with varying Coverage-Support Score Threshold

References

[1] Siliang Li, Bin Xu, and Tong Lee Chung. “Definition Extraction with LSTM Recurrent Neural Networks”. In: *CCL*. 2016.

[2] Roberto Navigli and Paola Velardi. “Learning Word-class Lattices for Definition and Hypernym Extraction”. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. ACL ’10. Uppsala, Sweden: Association for Computational Linguistics, 2010, pp. 1318–1327. URL: `http://dl.acm.org/citation.cfm?id=1858681.1858815`.