Language Understanding Systems
Second midterm report

Williams Rizzi
Master in Computer Science
University of Trento
williams.rizzi@studenti.unitn.it

Abstract

Modern information systems are able to store huge amounts of data. Is not anymore approachable to design an expert system to handle those data. Is necessary to change paradigm and, one possible solution is to start reasoning by patterns. To process data through patterns or templates a possible solution is to make use of Conditional Random Fields(CRF). In this way is possible to stop bothering about each possible variant of data and focus over patterns of interest. One possible application of CRF in Language Understanding systems is in concept sequence tagging. This work wants to investigate the feasibility of this approach and compare it with previous techniques.

1 Introduction

Discovering patterns underneath a set of sequences is a shared common problem and, one possible approach is by exploiting Conditional Random Fields (CRF). CRF uses a discriminative undirected probabilistic graphical model to encode known relationships in sequences. Being able to optimally correlate sequences of tokens allows a better comprehension of problems like concept sequence tagging. This work wants to address the CRF performances to concept sequence tagging and compare them with the results obtained with Finite State Transducers (FST).

In the following sections will be analysed the used dataset, discussed the approach to the problem and given evidences of the scored results. The results will then be compared with the results of the first midterm discussing the pros and cons of both approaches. And finally, will be drawn the conclusions and the possible further developments of this work.

2 Dataset

This section wants to tackle the dataset composition and its peculiarities. The dataset used to train the model comes from the movie domain. The corpora is already divided in train and test set. The train set is composed of 3338 sentences whereas the test set is composed of 1048 sentences. Each sentence is of an average length of 7 words with a standard deviation of 3 words. This dataset is presented in the form of a list of sentences and, for each word in each sentence is given the respective Part of Speech(PoS) (Martinez, 2012), Lemma (Gross, 1998) and IOB-tag (Tjong Kim Sang and De Meulder, 2003). In the rest of the work the words $word$ and $token$ will be used interchangeably.

2.1 IOB-tags

The concepts in the dataset follow the CoNLL IOB notation (Tjong Kim Sang and De Meulder, 2003) in which each concept is classified as the part of the multi-word span it belongs to, as begin (B), in (I) or out (O) of the concept span. Under this perspective the concept span is a proper solution to all those word like name and surname that need their direct follower or the previous term to be labeled correctly. The words $concept$ and IOB_tag will be used from now on interchangeably. For a more detailed overview of the dataset please refer to the midterm project, Section 2.

In the next section will be discussed the approach adopted to address the challenges of the dataset.

3 Approach

This work wants to setup a simple machinery to investigate CRF performances in the concept sequence tagging challenge . The CRF framework[1] allows the user to input datasets with customizable features, i.e. Lemmas and PoS, customizing the features are then composed via a user input template. The solution adopted is processes the data by: (i) loading and preprocessing, (ii) applying the selected feature template and, (iii) building of the CRF model. Is appropriate to say that the design of this system if compared with the one of the first midterm is way simpler in terms of coding skills but a lot more difficult in terms of feature engineering. Taking into consideration a simple sentence like: "$who\ plays\ nemo\ in\ finding\ nemo$"The input format of the framework is reported in the snippet 1. Is possible to add as many features as need,

Snippet 1 The format of the input is required to meet the following standard, always have the same amount of columns in the file and put the IOB-tag as last column. The columns are separated by tabs, the sentences are separated by blank lines and the unknown words are identified by an '#' character.

```
who      WP   who   O
plays    VVZ  play  O
nemo     NN   nemo  B-character.name
#        #    #     O
finding  VVG  find  B-movie.name
nemo     NN   nemo  I-movie.name
```

for the purpose of this work the feature vector is $< word, PoS, Lemma, IOB\_tag >$. The template format requires then the user to select how to compose them as shown in Snippet 2.
In the following subsections is described the different input sets used for the experimentation, starting from the baseline.

3.1 Tokens

The baseline of this investigation is the input vector composed by $< word, IOB\_tag >$. Is important to recall that this is simply the initial input over which the algorithm will apply the actual template specified by the user and, therefore is reasonable to say that this simple representation in some cases might be meaningful enough to score competitive results.

[1] http://taku910.github.io/crfpp/

Snippet 2 The format of the template needs follow a simpler standard. The U represents a feature treated as an unigram and the B tells the program that the defined unigrams need to be interpreted as bigrams. The'$\%x$' is the initial delimiter of a coordinate in the sentence represented as a matrix in Snippet 1. And, there can be any other letter as delimiters or whatever the user needs.

```
#Unigram
U:%x[0,0]
U:%x[1,2]a%x[2,2]
#Bigram
B
```

3.2 Lemmas

For what regards the first investigation of the capabilities of the CRF approach the Lemmas are used in place of the tokens as the input of the approach ending up with a feature vector as follows $< Lemma, IOB\_tag >$. The Lemmas tend to be interpreted as a slight generalisation of the bare words. Using lemmas in place of the tokens should help the inference motor better generalise over cases with similar words but different conjugations.

3.3 Part-of-speech tags

The second investigation of the capabilities of the CRF approach the Pos are used in place of the tokens as the input of the approach ending up with a input vector as follows $< Pos, IOB\_tag >$. This approach is not focusing anymore on the semantics of the words used in the sentences, instead is reasoning on the grammar and on the structure composed by sequences of Pos. This approach is particular, is trying to understand whether of not there is any correlation between the grammar structures used and the actual meaning of what it is meant to say.

In the next subsection is described the features set templates used for the construction of the models on top of the previously described different sort of inputs.

3.4 Feature construction

The feature construction is a very important step for this algorithm. Two kind of a approaches are tested in this experimentation, the first involves the use of windows to improve the aware of the current state of its neighbours. The second approach involves the use of unigram and bigrams in

the experiments to improve the feature awareness of its direct followers or followee. Now that are the described all the experimentations performed and is a bit clearer what one might expect to happen when adding a feature or another in the next subsection is presented the composition of all this techniques to try get the best out of the CRF algorithm.

3.5 The composition of the previous

As the last investigation of the capabilities of the CRF is tried to combine the previously described techniques in order to strengthen the overall performances. All the previously better performing template are mixed, the input format of the data is as follows $< word, Pos, Lemma, IOB\_tag >$. Is very interesting noticing that since the amount of different unigrams used to build this latter model are quite similar in the format but very different in the actual meaning is a good idea trying to enhance the differences between them for instance by changing the character separator or duplicating the rules that are considered more important in order to give them more chances to be learned properly by the statistical inferencer that build the model of the CRF.

In the next section will be presented the experimental results from the depicted techniques.

4 Results

The following results are computed using the CoNLL evaluation script, which scores the prediction in terms of accuracy, precision, recall and F1. The sizes from one to five have been tested as windows. In the tables is reported the accuracy and F1. In Table 1, is reported the results of the baseline algorithm, showing the high importance of the amount of information given with the approach with unigram or bigram. For all the reported approaches the CRF parameters are set to $cut\_off = 5$ and $CRF\_hiperparameter = 1.5$, those setting shown empirically to give best overall performances.

Table 1: Baseline

Window	N-Gram	Acc	F1
0	1	81.14%	33.51%
	2	88.03%	63.91%

In the Table 2, is possible to notice the very high impact of using the window technique constructing the feature vectors, especially if compared with the results in Table 1. The window value of Tables 2, 3 and 4 acts as a lookahead and a lookback function that shows what follows the token, or what precedes it. Hence, a window value of 1 means seeing the current value, the next and the previous one as features to predict the current label.

Table 2: Token

Window	N-Gram	Acc	F1
1	1	90.25%	58.42%
	2	91.54%	74.14%
2	1	92.40%	70.46%
	2	92.76%	78.30%
3	1	92.61%	73.55%
	2	92.83%	78.45%
4	1	92.41%	73.83%
	2	92.60%	77.78%
5	1	92.37%	73.81%
	2	92.72%	78.33%
6	1	92.31%	73.47%
	2	92.43%	77.55%

In the Table 3, is possible to notice the slight enhancement of the lemmas over the use of the bare tokens as in Table 2. Is interesting noticing that even if we expected the lemmas to better perform the tokens they result outperforming only in part of the configurations.

Table 3: Lemma

Window	N-Gram	Acc	F1
1	1	90.47%	58.99%
	2	91.88%	74.59%
2	1	92.37%	69.77%
	2	92.89%	77.97%
3	1	92.85%	73.91%
	2	92.82%	78.25%
4	1	92.68%	74.58%
	2	92.81%	78.76%
5	1	92.54%	73.83%
	2	92.40%	77.39%
6	1	92.37%	72.96%
	2	92.41%	78.02%

In the Table 4, is possible to notice that, as already mentioned in Section 3.3, the pure grammatical approach is not as good as the more semantic oriented approaches. Moreover, is very interesting to notice that it confirms the pattern evidenced also

on the other approaches in which almost always a bigger window and gram show better results.

Table 4: Pos tag

Window	N-Gram	Acc	F1
1	1	75.27%	16.26%
	2	78.12%	33.83%
2	1	78.38%	27.61%
	2	80.38%	44.11%
3	1	80.54%	33.70%
	2	81.57%	47.41%
4	1	80.98%	36.31%
	2	81.40%	46.71%
5	1	81.03%	36.54%
	2	81.44%	46.46%
6	1	80.74%	35.43%
	2	81.58%	46.82%

In the Table 5, is possible to notice the supremacy of the approach compared with the others.

Table 5: Mixed approach

N-Gram	Acc	F1
2	94.31%	82.36%

Overall, the results are solid in terms of trends and quite decent despite the fact that the parameters are tuned empirically. The mixed approach resulted as the best performing approach in this experimentation.

In the next section the scored result will be compared with the ones from the previous midterm

5 Discussion

The presented approaches show competitive results in both terms of accuracy and F1. Overall the amount of parameters to tune is manageable but the feature vector design is a bit of struggle and, following common design patterns like using windows can help a lot. Of course, the result depends a lot from the features and even in the luckiest cases is still need a well designed feature template to achieve satisfying results. For what regards the comparison between the two approaches, FST and CRF, as expected the CRF outperforms the FSTs with little or no tuning at all. Overall, the results confirm what pointed out in the Section 3. The Token and Lemma approaches perform similarly with slightly better results for the Lemma. The Pos tag approach performances are not that good but they get better with bigger windows and bigram. Finally, the mixed approach outperforms all the other techniques and also the FSTs of the first midterm. The power of the mixed approach lies in the fact that is able to exploit all the strengths of the previous approaches at the same time giving more weight to one or another feature accordingly.

In the next section will be presented the conclusions of the designed solution and, will be drawn the possible further developments of the approach.

6 Conclusions

The results from the two compared approaches are interesting, is particularly worth noticing that even if the result coming from this latter work are more promising than the ones from the previous approach are easier to score and therefore, might be more suitable for some applications. In particular as already mentioned in the previous work the amount of parameters to understand and set in an approach is an important discriminating factor when choosing it. Is worth to mention that might be, as said also in previous works conclusion, a good idea to improve the hyperparameters tuning through more methodical solutions like GridSearch or Evolutionary algorithms. Another important fact is that the CRF if not properly tuned tend to overfit the train set, might be a good option to adopt bigger datasets and use a train set, a test set and also an evaluation set.

References

Maurice Gross. 1998. Lemmatization of compound tenses in english. *Lingvisticae Investigationes* 22(1):71–122.

Angel R Martinez. 2012. Part-of-speech tagging. *Wiley Interdisciplinary Reviews: Computational Statistics* 4(1):107–113.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, pages 142–147.