A Survey of Named Entity Recognition Systems

Nealan Vettivelu

Abstract—This survey aims to compare results of several named entity recognition approaches using the Conference of Natural Language Learning 2003 dataset.The systems will be analysed in order to understand the benefits of certain features and as well strengths and weaknesses of approaches.

1 Introduction

Named entities are elements within a text such as a person's name, an organisation's name, and the name of a location. Named entity recognition (NER) is the process of classifying these terms within a document. By identifying these terms, we can provide more contextual information for a document resulting in an overall stronger understanding of the underlying concepts. Ratinov [1] demonstrates that the need for a complex model to determine NE extends beyond a local context and having prior knowledge is advantageous.

Even with a model that is able to capture prior probabilities, another major challenge that surrounds NER is that different languages require different features. By generalising rules, we aim to create a language independent model which can accurately conduct NER.However, this creates complexities due to the differences between languages both syntactically and semantically.Two overarching approaches have been used to counteract this difficulty – supervised and semi-supervised learning. This survey will look at the merits and sacrifices made by several methodologies in both approach types.

2 Approaches

The approaches looked at in this paper are varied from experimental to state-of-the art, in order to show the progression and various ideas that have occurred within this field. It is important to note that although some of these approaches may not be the strongest performers, their approaches towards NER were novel enough to provide a differing perspective from the norm.

2.1 Supervised Learning

Supervised learning is learning method in which all data has been classified previously. There are several considerations when looking at a supervised NER methodology. The first is the feature identification that is to identify the possible features that will be used. The second is feature reduction which looks at the reducing the feature set so that unnecessary features are not included with the model.Finally, the model or classifier used it itself another consideration when looking at an NER methodology. There are many approaches to take when looking at these methodologies, and this survey attempts to explore unique and novel approaches used for each of these considerations.

2.1.1 Feature Identification

Feature identification, looks at the features that have been identified by each approach and the importance the authors believe that they have towards the model. By incorporating non-sensical meaning to a model, they would likely be adding either coincidental data, or adding noise to the model. As such identifying features which have sufficient meaning is critical when creating a model.

Regardless of a models approach, orthography was one of the main components looked at in NER uses orthography as one of its primary features [2].By looking at the papers submitted for the CoNLL—2003 shared task, we can see that orthographic information was used in 12/16 systems. It is interesting to note that global case information was separated from this metric. This idea is validated by Akamatsu [3], who demonstrates that orthographic influences from a first language impact word recognition in English, specifically with an inappropriate capitalisation of words.

By primarily focusing on orthography rather than a more standardised feature set, Whitelaw and Patrick [4] have been able to show this importance as well. This approach notes the importance of capitalisation within a word and its relation to recognition performance. Furthermore, they highlight a difference between both English and German with relation to the use of capitalisation showing that restoring case information for German is much harder.

Two approaches were used for case restoration used by Whitelaw and Patrick. The first being a probabilistic approach in which the most common casing of a word was used. The issue with this approach was that common words such as 'new' would be incorrectly capitalised in 'New York'. The second approach used looked at an 'M-D trie' [5] in order to classify unseen words on the longest matchable prefix. Alternatively, by keeping both the unchanged and lower-cased version of a word as features this problem can be circumvented [6], although increasing the number of overall features.

Due to the importance of contextual information in both English and German [7] [8] the neighbourhood around a specific token are quite significant in classification. Both Mayfield [6] and Florian [9] highlight this importance through the sheer quantity of features used within their models, demonstrating this importance. By building upon the strictly orthographic approach shown by Whitelaw and Patrick, both Mayfield and Florian were able to capture additional information.

Another common trend was the use of part of speech (POS) information. It's importance has been recognised [10] and also has been a common feature amongst the approaches compared in this survey. Finally, the use of an n-gram feature has also been used in order to capture more contextual information about a token.

2.1.2 Feature Reduction

The next consideration when looking at a supervised NER method is the feature set used and its depth. In order to model a dataset well, an appropriate number of features are needed to ensure that there is enough precision and recall within the model whilst maintaining enough generality. An approach by Mayfield [6] looked at use of an extremely large feature set, which was then reduced using an SVM-lattice in order to ascertain the most relevant features. This method primarily focused on number of features and relied on the SVM-lattice to remove any unnecessary or weaker features. This approach was founded on the basis that with enough features, NER within a model can be accurately seen without the need for fine tuning. Examples of categories used include more common features such as the lemma to more niche ideas such as the word's position within a sentence. By creating a lattice structure, they were able to prevent impossible transitions much like Whitelaw's method suggesting a similarity between these methods, in removing redundant search paths. Much like Whitelaw, a HMM was used for calculations. A major benefit of their method is that SVM's are resistant to over training. As such, the use of SVM's to determine the best use features seems appropriate. By allowing the SVM to handle feature selection rather than the user, this mitigates the user's need for an expert understanding of the language being processed.

2.1.3 Classifiers

Another consideration that needs to be looked at is the classifier chosen, as different classifiers will look at data in differing ways. Classifiers include transformation-based learning, HMM, Robust risk management classifiers and maximum entropy classifiers to name a few.

One of the classifiers examined in this survey was a support vector machine (SVM). By using an SVM, Mayfield [6] could overcome the limitation presented by an excessive number of features. Due to an SVM creating a binary classification rather than a probabilistic one, we end up with an extremely large feature space that is both time consuming to create as well as resource intensive. Another classifier explored in this survey was the Hidden Markov Model (HMM). This model looks at prior probabilities of latent variables to determine the most likely class for the current token. Whitelaw and Patrick [4] show that using a HMM allows for an optimal sequence to be found when supplemented with additional dynamic programming practices. By understanding that natural text is sequential in nature, Ratinov [1] used Conditional Random Fields. The benefits of this model are through its ability to define a PDF over a sequence of observations rather than independently defining them. By using a two stage prediction model, they were able to trim the number of features required for successful classifications. We can say that although these models are different, a common theme amongst them is that feature reduction is used to save both time and space.

Looking at comparisons of classifiers [11], shows that not all classifiers are equal when it comes to the limitations each model has. By combining outputs of these classifiers, through either voting or stacking, much more rounded results can be gathered. These results can be further improved through the use of parameter or feature tuning.

An approach by Florian [9] shows the benefits of combining classification models in order to determine the best classification. This approach shows both the benefits and detriments caused by combining predictive models. By removing weaker performing classifiers, a stronger more accurate classifier can be bred. In order to combine the classifications from multiple classifiers, two approaches were looked at, a non-weighted vote and a weighted vote for each classifier. By using a weighted vote, the accuracy of each classifier for a given vote can be appropriately measured. Furthermore, the use of a partial credit votes to alternative classifications also allows for a deeper classification model. However, when looking at the German development data, we can see that the issues surrounding orthography presented by Whitelaw also affected this combination of classifiers as it relied heavily on capitalisation of words.

2.2 Semi-Supervised Learning

One of the major disadvantages of a completely supervised system is the need for labelled data, by using unlabelled data, we can overcome this problem. Unlike a Supervised learning system; a semi-supervised learning is a system that does training on both labelled and unlabelled data; as such the considerations in the system design are different from the considerations taken when looking at a supervised systemin order to account for these changes. Furthermore, the use of semi-supervised learning allows for training time of a model to be reduced [12].

2.2.1 Feature Selection

Due to the abundance of unlabelled data within a semi-supervised approach - a strong understanding of how to process data is required. As the unlabelled data will be a driving force behind the model the methodologies seen in the previous section will be insufficent. Due to this, the semi-supervised methods looked at in this survey are a Bagging-based model [12], Self-Learning Features [13] and two-stage prediction model [1].

The first semi-supervised method we will look at is a method that uses a bagging based approach [12]. This is a two-phased learning model in which a distant supervision generates a weakly labelled dictionary followed by an active learning ph ase which helps to refine the weakly labelled dictionary. This approach utilises multiple multi layered neural networks each given a different feature set from an NE dictionary. By aggregating the output via either weighted scoring or other means (seen also with Florian [9]), an output was created.

Clustering works on the assumption is that words with similar feature sets should be classed as the same named entity. As previously stated, semi-supervised learning models utilise both unlabelled and labelled data. By finding similarities between unlabelled data and labelled data we are able to create clusters, and as such create a larger feature set for the output classes. However, the approach to the use of clusters changes as shown by Qi [13] where clusters were used to group SLF distribution vectors so that they formed clusters, an approach very similar to k-means clustering. Clutsers were also explored by Ratinov [1] through the use of a binary tree very similar to the trie data structure proposed by Whitelaw.Both Qi and Ratinov's approach looked for similar contextual information and as such would place them into the same cluster when matched. This would allow for abstractions to be created and as such solve the issue of data sparsity that is inherit with natural language processing problems.

Much like with a bagging approach, the self-learning features approach suggested by Qi [13] which looks at predicted labels for a token over all instances within a corpus and then determining the most appropriate tag. This approach is like the bagging approach mentioned previously, due to the use of a multi layered neural network. This approach is beneficial when compared to supervised learning techniques due to its scalability. This is possible as the model works with features rather than examples.

Unlike previously mentioned models, the Qi's model is extremely robust due to its ability to predict any given feature as it uses a self-learning mechanism rather than relying on a predefined input dataset. By using the features of neighbouring words for unlabelled tokens in conjunction with a list of attributes they can more succinctly understand the type of entity that a word is. By loosely removing the need for feature selection and tying it to unsupervised techniques such as clusters and neural nets, the system proposed by Qi can overcome some of the limitations found in strictly supervised models.

2.2.2 Refining Results

To improve on the first epoch in a semi-supervised algorithm, the use of an update function is required. However, due to the vast abundance of unlabelled data it is not possible to base the error corrections strictly on the labelled data. One such approach to reduce the error rate in a model was shown by Lee [12]. Lee looked at the top 10% of most disagreed output, and manually tagged those results. By repeating this process they were able to steadily increase their F-score on a dataset. The benefit of this approach over the previously mentioned supervised approaches is that this approach uses a predefined dictionary to create a classification of text, rather than adapting to a labelled corpus it is given [14]. One of the disadvantages of this approach is that as noted by Lee, it requires a strong dictionary when beginning the recognition process.

Another approach used, seen by Qi [13] uses an averaging method over the unlabelled examples to smooth out errors. By doing this, infrequent mistakes are potentially corrected in the next iteration. By comparing the F1 scores provided [12] [13] , we can see that Qi's approach is able to plateau much earlier on, suggesting that the automated approach for refining the model is a far more effective and quicker. This may be due to the fact that the model is self-correcting and does not rely on user input. However, the trade off is that by using user input, the model is provided another level of depth. This is evident through the overall changes in score after several iterations of Lee's model (an increase of .02) compared to Qi's model(approximately 0.12).

3 Commonalities and Differences

Although the systems shown have used several different methodologies when looking at classification techniques, the use of a large feature set in a predictive model, the use of external knowledge and finally the use of a dynamic programming algorithm have all had an overarching similarity across the systems shown within this survey.

3.1 Preprocessing

3.1.1 Lemmatisation

For each system except for the first made by Whitelaw, the lemmatisation of each word had been used. Furthermore, when the lemmatisation was looked at another common feature that was used was the surrounding words. This common approach of using surrounding words shows that there is a strong consensus that local contextual information provides meaning for a word.

3.1.2 Tagging Schema

The tagging schema's varied amongst systems. The tagging methodolgies found were BIO,IOBES and BILOU tagging system for chunks of text. Each system, represents a token in a sequence in different ways [15]. Understanding these differences allows for a different interpretation on not only the document but the corpus itself . Konkol [16] shows that for an English corpus the use of the IEO-1 and IEO-2 tag sets were the strongest performers dependent for a CRF and Maximum Entropy (ME) model respectively suggesting that there is a possibility for improvement with all the aforementioned approaches through this different tag set.

3.1.3 Gazetteers

Gazetteers are dictionaries that have been created before and during the modelling of a text manually to allow for certain tokens to be more readily identified despite their context. All of the aforementioned models have made use of gazetteers in order to allow for a stronger classification model.Whilst most models used gazetteers as suplimental knowledge, Ratinov [1] used gazetteers extensively throughout the classification process as it was noted that features found within a Wikipedia article could contain several pieces of information found outside the main text such as Microsoft having the category *Companies listed on NASDAQ: Cloud computing vendors;*. Furthermore Ratinov, noted that the use of several gazetteers from various sources allows for a more robust model due to the natural tendency of gazetteers having a high accuracy. The reason that a model cannot solely exist from the use of gazetteers was due to the lack of recall they provide.

3.2 Updating the Model

3.2.1 Use of Dynamic Programming

One of the major barrier with NLP tasks, is the resource and time required. In order to alleviate some of these stresses on a system dynamic programming has been a major trend amongst these approaches.The Viterbi algorithm, was an algorithm that appeared often amongst these papers. By using the Viterbi algorithm issues regarding both space and time complexity become much more trivial due to the algorithm reducing the amount of space required to create an output.

3.2.2 Aggregation of Classifiers

The aggregation of classifiers allows smaller variations in predictions to be accounted for due to the use of a voting system. This was a novel approach used by Florian [9] to classify data using a supervised approach. However, this is common practice within the semi-supervised methodologies due to the popular use of bootstrapping, in which multiple neural networks are used in order to develop a solution by starting out with random feature sets, rather than the same feature set as seen in Florian's paper.

4 Conclusion

In conclusion, we can see that both supervised and semi-supervised approaches have their own merits and cons. It should be noted that the semi-supervised approach is able to take advantage of the abundance of unlabelled text which is available whilst a supervised approach is not. Semi-supervised learning systems still encounter the problem of determining whether a predicted term is accurate, and as such still require extensive use of external knowledge.

References

[1] Lev Ratinov and Dan Roth. Design Challenges and Misconceptions in Named Entity Recognition *. pages 147–155.

[2] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 -*, volume 4, pages 142–147, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[3] Nobuhiko Akamatsu. The effects of first language orthographic features on word recognition processing in English as a second language. *Reading and Writing*, 11(4):381–403, 1999.

[4] Casey Whitelaw and Jon Patrick. Named Entity Recognition Using a Character-based Probabilistic Approach. *Australian Conference on Artificial Intelligence*, pages 910–921, 2003.

[5] Donald Ervin Knuth. *The art of computer programming*. Addison-Wesley Pub. Co, 1973.

[6] James Mayfield, Paul Mcnamee, and Christine Piatko. Named Entity Recognition using Hundreds of Thousands of Features.

[7] Christoper D Manning and Hinrich Schutze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[8] Citation Shieber and Stuart. Evidence against the context-freeness of natural. *Linguistics and Philosophy*, 8:333–343, 1985.

[9] Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. Named Entity Recognition through Classifier Combination.

[10] Ruslan Mitkov. The Oxford Handbook of Computational Linguistics - Google Books, 2005.

[11] Jakub Zavrel, Sven Degroeve, Anne Kool, Walter Daelemans, and Kristiina Jokinen. Diverse Classifiers for NLP Disambiguation Tasks Comparison, Optimization, Combination, and Evolution.

[12] Sunghee Lee, Yeongkil Song, Maengsik Choi, and Harksoo Kim. Bagging-based active learning model for named entity recognition with distant supervision. In *2016 International Conference on Big Data and Smart Computing (BigComp)*, pages 321–324. IEEE, jan 2016.

[13] Yanjun Qi, Pavel Kuksa, Ronan Collobert, Kunihiko Sadamasa, Koray Kavukcuoglu, and Jason Weston. Semi-Supervised Sequence Labeling with Self-Learned Features. In *2009 Ninth IEEE International Conference on Data Mining*, pages 428–437. IEEE, dec 2009.

[14] Kyoungnam Ha, Sungzoon Cho, and Douglas MacLachlan. Response models based on bagging neural networks. *Journal of Interactive Marketing*, 19(1):17–30, jan 2005.

[15] Vijay Krishnan and Vignesh Ganapathy. Named Entity Recognition. 2005.

[16] Michal Konkol and Miloslav Konopík. Segment representations in named entity recognition. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9302:61–70, 2015.