Named Entity Recognition - A Literature Survey

Anmol Singh Suag University of Massachusetts - Amherst 21st March, 2018

asuag@umass.edu

1. Introduction

Named Entity Recognition (NER) is one of the core tasks of Natural Language Processing (NLP) and a first step for Information Extraction. A 'named entity' is token or sequence of tokens in a language that are proper names of a person, location or organisation. As dates, numbers and times, etc. are important to Information Extraction, the term has been extended to incorporate temporal expressions as well as numerical expressions. The term was coined for the Sixth Message Understanding Conference (MUC-6) whose aim was to promote research on extraction of structured data from unstructured text. Extracted entities would serve as an important first step for variety of other tasks like question answering, linking entities to knowledge sources, etc.

NER is a challenging problem as there is hardly any restriction on what could be a name. Moreover, if a name consists of multiple tokens, a subset of these tokens is often used in subsequent text to refer to the same named entity, hence raising a need for relation extraction. In the early years, the research focussed majorly on hand-crafted rule based algorithms, as well as utilising heavy pre-processing and external knowledge sources. Task and language specific feature modeling hindered generalisations of earlier NER systems. With the progress of research in NER, machine learning techniques were employed with an aim to minimise dependency on hand-crafted features, but researchers were tempted to optimise the performance of their system for a specific benchmark. In the recent years, the research is focused on using variations of deep learning models, as well as optimisations to speed up the otherwise expensive training for NER.

The survey aims to provide a short history of the progress of NER research over the last two decades. We would discuss the evolution of various learning techniques, features and external knowledge sources, evaluation of NER models and many state-of-the-art systems of their times. We would look deeply into the structure of recent neural network based models and compare their architecture, as well as, performance.

2. A Brief History

One of the first few works in NER was presented by Lisa F. Rau (1991)[20] as a system to extract and recognize company names from using heuristic and hand-crafted rules. Research in NER accelerated after MUC-6 (R. Grishman & Sundheim 1996 [10]) and has been flourishing ever since with bolstering scientific events like CoNLL (E. Tjong Kim Sang 2002; E. Tjong Kim Sang & De Meulder 2003[7]), ACE (G. Doddington et al., 2004[11]), HAREM (D. Santos et al. 2006[1]) and ACL workshops.

Early majority research in NER had been for English language, but CoNLL 2002, 2003 shared tasks included data-sets for German, Spanish, Dutch and Portuguese, therefore promoting research for NER in other languages and language independence of models. Although numerous papers focussed on individual languages like French (G. Perasis et al.,2001; Poibeau 2003[15]),

Italian (W. Black et al. 1998[21]), Greek (S. Boutsis et al. 2000), Chinese(S. Yu et al. 1998), later researches advocated for language independence. Florian et al. 2003 obtained the best score on German by combining output of four diverse classifiers. Qi et al. (2009)[13] improved this score by using a neural network with unsupervised learning on huge unlabeled corpus. Carreras et al. (2002)[12] obtained state-of-the-art on both Dutch and Spanish by combining several fixed-depth decision trees. There is still active research in south-asian languages like Urdu, Hindi etc. owing to their non-capitalised and phonetic complexities (K Riaz, 2010[29]).

Machine Learning techniques and statistical approaches became a major area of research on NER, especially after Lafferty et al. proposed Conditional Random Field in 2001. The famous state-of-the-art systems are Stanford NER (Finkel et al., 2005) and UIUC NER (Ratinov and Roth, 2009). Recent works on NER have started to focus on multilingual NER and NER on short text, eg Twitter. Approaches have been dominated by applying CRF, SVM or perceptron models to hand-crafted features (Ratinov and Roth, 2009; Passos et al., 2014; Luo et al., 2015). Collobert et al. (2011) proposed an effective neural network model that requires little feature engineering and learns important features from word embeddings trained on large quantities of unlabelled text. This was made possible by advancements in unsupervised learning of word embeddings on massive amounts of data (Collobert and Weston, 2008; Mikolov et al., 2013) and neural network training algorithms permitting deep architectures (Rumelhart et al., 1986). Ratinov and Roth (2009) used non-local features, a gazetteer extracted from Wikipedia, and Brown-cluster-like word representations, and achieved an F1 score of 90.80 on CoNLL- 2003. Lin and Wu (2009) surpassed them without using a gazetteer by instead using phrase features obtained by performing k-means clustering over a private database of search engine query logs.

Passos et al. (2014) obtained almost the same performance using only public data by training phrase vectors in their lexicon-infused skip-gram model. In order to combat the problem of sparse features, Suzuki et al. (2011) employed large-scale unlabelled data to perform feature reduction and achieved an F1 score of 91.02 on CoNLL-2003, which is the current state of the art for systems without external knowledge.

Petasis et al. (2000) used a feed-forward neural network with one hidden layer on NER and achieved state-of-the-art results on the MUC-6 dataset. Their approach used only POS tag and gazetteer tags for each word, with no word embeddings. Collobert et al. (2011b) presented SENNA, which employs a deep FFNN and word embeddings to achieve near state of the art results on POS tagging, chunking, NER, and SRL.

With widespread success of BLSTMs for NER, various architectures with a combination of CRFs, CNNs and BLSTMs were presented around 2015. Huang et al. (2015) used a BLSTM for the POS-tagging, chunking, and NER tasks, but they employed heavy feature engineering instead of using a CNN to automatically extract character-level features. Labeau et al. (2015) used a BRNN with character-level CNNs to perform German POS- tagging. Ling et al. (2015) used both word- and character-level BLSTMs to establish the current state of the art for English POS tagging. Ma and Hovy (2016)[4] introduce a novel neural network architecture that benefits from both word- and character-level representations automatically, by using combination of bidirectional LSTM, CNN and CRF. Chiu et al.(2016)[9] present a novel neural network architecture which is a hybrid model of bidirectional LSTMs and CNNs that learns both character and word-level features automatically, hence eliminating the need for most feature engineering. Luo et al. (2015)[26] build a joint model for Entity Recognition and Disambiguation (ERD).

Lastly, research to decrease training time of neural networks is in progress and Strubell et al. (2017)[38] propose a faster alternative to Bi-LSTMS for NER and call it Iterated Dilated Convolutional

Neural Networks (ID-CNNs), which have better capacity than traditional CNNs for large context and structured prediction.

3. Evolution of Learning Models and Techniques

NER is typically viewed as a sequence labeling task. Many statistical hidden state sequence models have been employed for Information Extraction. These include Hidden Markov Models (HMMs) (Leek, 1997[27]; Freitag and McCallum, 1999[22]) , Conditional Markov Models (CMMs) (Borthwick, 1999[28]) and Conditional Random Fields (CRFs) (Lafferty et al., 2001 [23]). All these model encode the Markov Property that a state at a particular position in a sequence can only depend on a small local window. It is a property that allows tractable computation, a basis for many inference algorithms like Viterbi, Clique Calibration etc. As non-local structure is important for NER, Finkel et al. 2005[33] used Gibbs sampling, a simple Monte Carlo Method to perform inference, with an existing CRF based information extraction system based on Lafferty et al., 2001 [23]. This enabled incorporation of non-local structure and presence tractable inference. The technique resulted in 9% error reduction over then state-of-the-art systems.

3.1 Heavy Hand-crafted Features

Initial approaches incorporated heavy use of hand-crafted features and external knowledge sources. Florain et al. (2003) had the best system for CoNLL 2003 with an F1 score of 88.76. They used a combination of many machine learning classifiers and, various hand-engineered features and a large gazetteer. Chieu (2003) [30] scored an F1 of 86.84 and also used an external gazetteer and various engineered features. Ando and Zhang (2005)[16] scored F1 of 89.31 but had a semi-supervised approach with lesser dependency on external features.

Semi-supervised learning involves a small degree of supervision, such as seeds, for starting the learning process. Semi-supervised learning approaches for parameter sharing were used by Ando and Zhang (2005)[16], Suzuki ahd Isozaki(2008)[24] where models were jointly trained with a multi-task approach. It is seen that unlabeled text can be used to improve the performance of NER systems greatly using word clusters, a technique that goes back to Brown et al. 1992[14]. Ratinov and Roth, 2009[3] present a simple model that uses expressive features to achieve a new state-of-the-art NER model. They experiment with various non-local features, external knowledge sources and inference methods. Their experiments bolster that word class models learned on unlabeled text can be an alternative to traditional semi-supervised techniques.

3.2 Towards Language Independence

Collobert et al. 2011[2] impacted the research in NER in a major way. Minimising the pre-processing on input features, they train a multi-layer neural network architecture in an end-to-end fashion. The architecture takes the input sequence and learns several layers of feature extraction that process inputs. Word lookup tables of supervised networks are initialised with embeddings computed by language models. The supervised training is able to modify the contents of the look-up tables. Once the language models are trained, multiple experiments can be performed on the supervised networks in a short span of time. This approach is linked to deep learning techniques of Hinton et al.(2006)[45] , Bengio et al.(2007)[32] and Weston et al. (2008)[31]. Collobert et al. tried to excel on multiple benchmarks by avoiding task specific feature engineering, a goal that has been kept in mind by the research community ever since. They argued that task-specific benchmarks are indirect measurements of how good the internal representations discovered by a model are and the internal representations should aim for generalisability over multiple NLP tasks. Their

model was able to reach good performance in most NLP tasks by transferring the intermediate representations learnt on large unlabeled datasets.

Mikolov et al. 2013[35] proposed two simple log-linear language models, the CBOW and Skip-Gram model, which are simplifications of neural language models and can be trained efficiently on large amounts of data. It is possible to train a Skip-gram model over a billion tokens with a single machine in half a day. These embeddings can be trained on phrases as well as words, allowing for finer granularity. Mikolov et al. use phrase building criterion based on pointwise mutual information of bigrams. Passos et al. 2014[6] instead consider candidate trigrams for all pair of bigrams to extend these embeddings to model phrases as well as tokens. Inspired by Ratinov and Roth, 2009[3] and Zhang and Johnson (2003), their baseline architecture is a stacked linear-chain CRF system. They train two CRFs, where the second CRF can condition the features of data and predictions made by the first CRF. The model incorporates lexicons of months, days, books, person names, events, organisations, films, etc. and features based on Brown clusters. Passos et al. argues that in NLP tasks like NER, syntax alone is not enough to build a high performance system and some external source is definitely required. They argue that in most state-of-the art systems for NER, this knowledge comes from domain specific lexicons and word representations that capture their syntactic and semantic behaviour. They argue that although attempts have been made to use other word representations from training neural language models, their performance is worse than the systems that use Brown clusters like Turian et al.(2010)[39]. Although neural language models are more scalable than Brown clusters, they are expensive to train.

Santos et al. (2015)[1] propose a language independent NER system that used automatically learnt features only. Their approach is based on CharWNN deep neural network that uses character-level and word-level representations to perform sequential classification. CharWNN extends Collobert et al. (2011) and adds a convolutional layer to extract character-level representations.

3.3 Joint Modeling of various NLP Tasks

Most existing NLP tasks like POS tagging, chunking, NER, entity linking and parsing had been handled separately in most previous researches. This decoupled nature of modeling different tasks separately hinders one task to take advantage of the information of other. A decoupled approach even poses a risk of inconsistent outputs from various tasks on the same data. An interest in joint modeling of various NLP tasks has been around. Joint optimisation models have been investigated by using Dynamic CRF (McCallum et al. 2003)[25] that aimed to conduct POS tagging and Chunking together. Finkel and Manning (2009[34]) modelled a parsing and NER task together. Sil (2013)[36] proposed a joint NER and linking task by leveraging existing NER systems and Freebase, but left the linking algorithm to make the final decisions, essentially becoming a re-ranking model. Luo et al. (2015)[26] build a joint model for Entity Recognition and Disambiguation (ERD). The goal of ERD is to extract named entities in text and link extracted names to a knowledge base like wikipedia. Where Sil uses an existing state-of-the-art NER model, Luo et al. model the two tasks jointly from the training phase itself. Their model is more powerful in capturing mutual dependencies.

Luo et al. propose Joint Entity Recognition and Linking (JERL) to jointly model NER and linking tasks and capture the mutual dependency between them. It allows for using the information from one task to improve the performance of the other. It is the first model to optimise NER and linking tasks together completely. Although expensive, joint optimization is a promising direction as it is closer to how humans process text information. JERL is a probabilistic graphical

model that uses many varied features like dictionaries, Brown clusters, WordNet clusters, character n-grams and unigram/bigram for word, lowercase words, word shapes, chunks and characters.

3.4 Neural Networks with Bi-directional LSTMs

NER has been a challenging task that has always needed large amounts of knowledge in the form of feature engineering and lexicons to achieve high performance. Neural Networks have been used for for automatic feature learning for variable length inputs. A well-studied solution for a neural network to process variable length input and have long term memory is RNN (Goller and Kuchler, 1996)[43]. The LSTM unit with a forget gate allows highly non-trivial long-distance dependencies to be easily learnt (Gers et al., 2000)[17]. A bi-directional LSTM model can take into account an effectively infinite context on both sides and eliminate the problem of incomplete text in feed-forward models (Graves et al.,2013)[18]. CNNs have also been investigated for modeling character-level information. Santos et al. (2015) successfully employed CNNs to extract character-level features for NER. Collobert et al.[2] used CNNs for semantic role labeling, however the effectiveness of character-level CNNs had not been evaluated for English NER. Collobert et al. (2011)[2] has many limitations as it uses a simple feed-forward neural network that restricts the context size to a fixed window around a word and doesn't leverage on long-distance relation between words. Moreover, by depending solely on word embeddings, it is unable to exploit character level features such as prefix and suffix.

Huang et al. (2015)[37] propose a variety of LSTM based models for sequence tagging. Their work is the first to apply bidirectional LSTM CRF model to NLP benchmark sequence tagging. The BLSTM component efficiently captures both past and future context and the CRF layer captures the sentence level tag information. It makes sense to use Bidirectional LSTM as proposed by (Graves et al., 2013)[18] as we have access to both past and future input features for a given time-step. Furthermore, there are two different ways to make use of neighbor tag information in predicting current tags. The first is to predict distribution of tags for each timestep and then use a beam-like decoding to find optimal tag sequences and the second is to focus on sentence-level instead of individual positions. The former approach encompasses works of Maximum Entropy Classifier (Ratnaparkhi, 1996[42]), Maximum Entropy Markov Models (MEMMs)(McCallum et al., 2000) and the latter approach includes CRF models. Huang et al. use CRF with Bi-LSTM to boost tagging accuracy and achieve state-of-the-art performance at their time.

Chiu et al.(2016)[9] present a novel neural network architecture which is a hybrid model of bidirectional LSTMs and CNNs that learns both character and word-level features automatically, hence eliminating the need for most feature engineering. The model derives from Collobert et al, (2011b)[2] where lookup tables transform discrete features such as words and characters into continuous vector representations. Instead of feed-forward neural network used by Collobert et al., Chiu et al. use BLSTM and to induce character-level features. CNNs have been successfully applied to Spanish and Portuguese NER (Santos et al., 2015)[1] and German POS-tagging (Labeau et al. 2015)[19]. The training involves a carefully tuned dropout hyperparameter that is responsible for the state-of-the-art performance. Character level Bi-LSTMs which were recently proposed by Ling et al. (2015)[40] for POS tagging weren't found to perform significantly better than CNNs and were more computationally expensive. While using BLSTMs instead of CNNs allows extraction of more sophisticated character-level features, Chiu et al. found that for NER it did not perform significantly better than CNNs and was substantially more computationally expensive to train. Moreover, Huang et al. (2015)[37] used a BLSTM for the POS-tagging, chunking, and NER tasks, but they employed

heavy feature engineering instead of using a CNN to automatically extract character- level features.

Lample et al. (2016)[5] introduced two novel neural architectures, one based on bidirectional LSTMs and conditional random fields (CRF) and the other based on constructing and labelling segments using transition-based approach derived from shift-reduce parsers. Carefully constructed orthographic features and language-specific knowledge resources like gazetteers is itself a challenge owing to unrestricted nature of Named Entities. Alternatively, unsupervised learning strategy in an attempt for better generalisation, ends up using unsupervised features to augment hand-crafted features and specialised-knowledge resources (Collobert et al., 2011; Lin and Wu, 2009; Turian et al., 2010). Lample et al. doesn't use any language-specific resources, but only a small amount of supervised training data and unlabelled corpora. Token-level evidence for being a name depends on orthogonal evidence, what does the word being tagged as a name look like, as well as on distributional evidence, where does the word occur in corpus. Character-based word representation model (Ling et al.,2015b)[40] is used to capture orthographic sensitivity and these representations are combined with distributional representations (Mikolov et al. 2013b)[35] to capture distributional sensitivity. Training includes dropout to learn both sources of evidence. The LSTM-CRF model is able to achieve state-of-the-art NER performance in Dutch, German, Spanish and very close to English.

Ma and Hovy (2016)[4] introduce a novel neural network architecture that benefits from both word- and character-level representations automatically, by using combination of bidirectional LSTM, CNN and CRF. Their system is actually end-to-end, and requires no feature engineering or data preprocessing, hence making it feasible for variety of sequence labeling tasks. Various nonlinear neural network models discussed previously have utilized distributed representations as inputs and have used these to augment, rather than replace, hand-crafted features (e.g. word spelling and capitalization patterns). Their performance drops rapidly when the models solely depend on neural embeddings. For example, English POS taggers benefit from carefully crafted word spelling features like orthographic features and external resources such as gazetteers are widely used in NER. Such task-specific knowledge is expensive to create (Ma and Xia, 2014)[4], making sequence labeling models difficult to adapt to new tasks or new domains. Ma and Hovy use convolutional neural networks (CNNs) (LeCun et al., 1989)[44] to encode character-level information of a word into its character-level representation. Then the character-level and word-level representations are combined and fed into a bi-directional LSTM (BLSTM) to model context information of each word. On top of BLSTM, a sequential CRF is used to jointly decode labels for the entire sentence. Dropout layers are used on both the input and output vectors of BLSTM. It is shown that that using dropout significantly improves the performance.

3.5 Comparing the BLSTM Models

Huang et al. used BLSTM for word-level representations and CRF for joint label decoding, which is similar to our Ma and Hovy. There are two differences between these two models: Firstly, Huang et al. did not employ CNNs to model character-level information. Secondly, they augmented their model with hand-engineered features to improve their performance, and therefore making their model not an end-to-end system. Chiu et al. proposed a hybrid of BLSTM and CNNs to model character and word-level representations, which is similar to the first two layers in Ma and Hovy. Chiu et al. evaluated their model on NER and achieved competitive performance. Ma and Hovy's model mainly differ from this model by using CRF for joint decoding. Chiu et al.'s model is not truly end-to-end too as it uses external knowledge sources such as character-type, capitalization and lexicon features, as well as some data pre-processing specifically for NER. Lample et al. (2016) proposed a BLSTM-

CRF model for NER, which utilizes BLSTM to model both the character-level and word-level information, and use data preprocessing the same as Chiu et al. Ma and Hovy use CNN to model character-level information, achieving better NER performance and that too without using any data preprocessing

3.6 Towards faster Named Entity Recognition

As of today many applications run basic NLP on the entire web and huge data traffic, therefore faster methods are a need of the hour to save time and energy costs. Advances in GPU hardware have led to the rise of bi-directional LSTMs as the main method to obtain per-token vector representations that serve as an input to labeling tasks such as NER. Although expressive and accurate, these models fail to fully utilise GPU parallelism and hence limiting their computational efficiency.

Strubell et al. (2017)[38] propose a faster alternative to Bi-LSTMS for NER and call it Iterated Dilated Convolutional Neural Networks (ID-CNNs), which have better capacity than traditional CNNs for large context and structured prediction. Unlike LSTMs whose sequential processing on sentences of length N requires O(N) time even during parallelism, ID-CNNs work by enabling fixed-depth convolutions to run in parallel across full documents. The paper proposes a distinct combination of network structure, parameter sharing and training procedures that have shown 14-20x test-time speedups while retaining accuracy comparable to the Bi-LSTM-CRF. Dilated Convolutions were proposed by Yu and Kotlun, 2016 and like CNN layers, they work on a sliding window of context over the sequence. Unlike conventional convolutions, the context need not be consecutive and the dilated window skips over every dilation width d inputs. By stacking layers of dilated convolutions of exponentially increasing dilation width, the size of the effective input width is increased to accommodate the full length of most sequences using only a few layers. This iterated dilated CNN architecture (ID-CNN) repeatedly applies the same block of dilated convolutions to token-wise representations. This parameter sharing prevents overfitting and also provides opportunities to inject supervision on intermediate activations of the network. Then Viterbi inference is used to predict each token's label. ID-CNN can even be used to feed a CRF model. Dropout is used to prevent overfitting. Moreover, the version of dropout normally used in practice has the undesirable property that the randomized predictor used at train time varies from the fixed one used at test time. Ma et al. (2017) present dropout with expectation-linear regularization, which explicitly regularizes these two predictors to behave in the same way.

4. Tagging Schemes

The task of NER is to assign a named entity label to words in a sentence. A named entity could span multiple tokens and tokens in a sentence are generally represented in IOB format (Inside, Outside and Beginning) depending on their position in the named entity. Lample et al. use IOBES tagging scheme that encodes information about singleton entities (S) and explicitly marks the end of named entities (E). Using this scheme, tagging a word as I-label with high confidence narrows down the the choices for the subsequent words to be I-label or E-label. Whereas IOB scheme is only capable of determining that the subsequent word can't be interior of another label.

Chiu et al. (2016)[9] , Ma and Hovy (2016)[4] and Collobert et al. (2011)[2] also use BIOES tagging scheme as it is the most expressive. Passos et al. (2014)[6] use the BILOU encoding, where each token can either BEGIN an entity, be INSIDE an entity, be the LAST token in an entity, be OUTSIDE an entity, or be the single UNIQUE token in an entity.

Ratinov and Roth, (2009)[3] argue that the BIO scheme suggests to learn classifiers that identify

the Beginning, the Inside and the Outside of the text segments. The BILOU scheme suggests to learn classifiers that identify the Beginning, the Inside and the Last tokens of multi-token chunks as well as Unit-length chunks. The BILOU scheme allows to learn a more expressive model with only a small increase in the number of parameters to be learned. Experiments by Dai et al., 2015 have also shown that using a more expensive tagging scheme like IOBES improves model performance marginally, but Lample et al. found that it increases only minutely.

5. Model Evaluations

Most NER models report their performance on CoNLL 2002 (Tjong Kim Sang)[7] , CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003), Ontonotes (Weischedel et al., 2011)[8] and MUC-7 datasets. The CoNLL 2003 dataset has approximately 320k tokens, divided into 220k tokens for training, 55k tokens for development, and 50k tokens for testing. While the training and development sets are quite similar, the test set is substantially different, and performance on it depends strongly on how much external knowledge the systems have. The CoNLL dataset has four entity types: Person, Location, Organisation and Miscellaneous. The Ontonotes dataset is substantially larger: it has 1.6M tokens total, with 1.4M for training, 100K for development, and 130k for testing. It also has eighteen entity types, a much larger set than the CoNLL dataset, including works of art, dates, cardinal numbers, languages, and events.

The performance of NER systems is commonly measured in terms of precision, recall, and F1 on the sets of entities in the ground truth and returned by the system.

6. Performance

We would now discuss the performance statistics of various models discussed in this survey. The datasets are English unless otherwise specified. Finkel et al. (2005)[33] used Gibbs sampling with a CRF model and reported F1 score of 86.72 on CoNLL 2003 dataset. Ratinov and Roth (2009)[3] developed a publically available NER system with an F1 score of 90.8 on CoNLL 2003. Collobert et al. (2011)[2] ended up using language models and a gazetteer to reach an F1 score of 89.59 on CoNLL 2003. Using Lexicon infused embeddings and lexicons as features, Passos et al. (2014)[6] scored an F1 of 90.90 on CoNLL 2003, which matched Lin and Wu (2009)[41] but without using massive private data. Their Skip-gram variant scored an F1 of 82.30 on OntoNotes 5.0, which was the state-of-the-art then. Santos et al. (2015)[1] achieved an increment of 0.8 F1 points on Spanish CoNLL 2002 dataset. Luo et al. (2015) used JERL to report the highest F1 score of 91.2 on CoNLL 2003 task, but their model uses a lot of hand-engineered features including POS tags, Brown clusters, WordNet clusters as well as external knowledge basis like Freebase and Wikipedia. Lample et al. (2016)[5] score an F1 of 90.94 using their LSTM-CRF model on English , 78.6 on German, 81.74 in Dutch and 85.75 in Spanish. Being the state-of-the-art in Spanish and German, the LSTM-CRF model is close to state-of-the-art models for English and Dutch that use external labelled data. Gillick et al.,2015 score an F1 of 82.84 on Dutch. Huang et al. (2015) achieve the best F1 score of 90.10 with both Senna embedding and gazetteer features. Strubell et al. (2017)[38] propose a model that is incredibly faster than contemporary models with performance matching, but not surpassing them. Their ID-CNN-CRF model outperforms the Bi-LSTM-CRF model by 0.11 points of F1 on average.

In the race for the best F1 score on CoNLL 2003, Ma and Hovy (2016)[4] achieve the best F1 score of 91.21 with their truly end-to-end system. Chiu et al. (2015)[9] had reported their the F1 score of 90.77 on CoNLL 2003 and the same has been reported in most contemporary papers. But, there was a revision to the paper in 2016 and they reported an increased score of 91.62 on CoNLL 2003 and 86.28 on OntoNotes 5.0 which is the highest

performance until recently, to the best of our knowledge.

7. Conclusion

In this literature survey we have provided a brief overview of the direction of research in Named Entity Recognition task of NLP. We have covered the majority of research checkpoints and seen how the research shifts from pure rule-based , hand-crafted algorithms to automated feature learning techniques. We have observed that although language independence and multi-task modeling is the aim solving NLP tasks, there are temptations to incorporate external sources and task-specific features to compete on the evaluation tests. Most recent research have used combinations of BLSTMs, CNNs and CRFs and efforts for optimising the training time whilst maintaining performance are in progress.

We feel that techniques for multi-task optimisations, complete language independence and decreasing training time for NER as well as other NLP tasks would be given more importance in the NLP research community.

References

[1] [Santos and Guimara˜es2015] Cicero Nogueira dos Santos and Victor Guimara˜es. 2015. Boosting named entity recognition with neural character embeddings. *arXiv preprint arXiv:1505.05008*.

[2] [Collobert et al.2011] Ronan Collobert, Jason Weston, Le´on Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

[3] [Ratinov and Roth 2009] Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics.

[4] Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, page 10641074.

[5] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL*.
[6] Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. In *CoNLL*.
[7] 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*, pages 142–147. Association for Computational Linguistics.
[8] Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: the 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60.
[9] Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.
[10] Grishman, Ralph, and Beth Sundheim. "Message understanding conference-6: A brief history." *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*. Vol. 1. 1996.
[11] Doddington, George R., et al. "The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation." *LREC*. Vol. 2. 2004.
[12] Carreras, Xavier, et al. "FreeLing: An Open-Source Suite of Language Analyzers." *LREC*. 2004.
[13] [Qi et al.2009] Yanjun Qi, Ronan Collobert, Pavel Kuksa, Koray Kavukcuoglu, and Jason Weston. 2009. Combining labeled and unlabeled data with word-class distribution learning. In Proceedings of the 18th ACM conference on Information and knowledge management, pages 1737–1740. ACM.
[14] Brown, P.F., Desouza, P.V., Mercer, R.L., Pietra, V.J.D. and Lai, J.C., 1992. Class-based n-gram models of natural language. *Computational linguistics*, *18*(4), pp.467-479.
[15] Poibeau, Thierry. "The multilingual named entity recognition framework." *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 2*. Association for Computational Linguistics, 2003.
[16] Ando, Rie Kubota, and Tong Zhang. "A framework for learning predictive structures from multiple tasks and unlabeled data." *Journal of Machine Learning Research* 6.Nov (2005): 1817-1853.
[17] Gers, Felix A., and Jürgen Schmidhuber. "Recurrent nets that time and count." In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, vol. 3, pp. 189-194. IEEE, 2000.
[18]Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks." *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013.
[19]Labeau, Matthieu, Kevin Löser, and Alexandre Allauzen. "Non-lexical neural architecture for fine-grained POS tagging." *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015.
[20]Rau, Lisa F. "Extracting company names from text." *Artificial Intelligence Applications, 1991. Proceedings., Seventh IEEE Conference on*. Vol. 1. IEEE, 1991.
[21]Hair, J.F., Black, W.C., Babin, B.J., Anderson, R.E. and Tatham, R.L., 1998. *Multivariate data analysis* (Vol. 5, No. 3, pp. 207-219). Upper Saddle River, NJ: Prentice hall.
[22]Freitag, Dayne, and Andrew McCallum. "Information extraction with HMMs and shrinkage." *Proceedings of the AAAI-99 workshop on machine learning for information extraction*. 1999.

[23]Lafferty, John, Andrew McCallum, and Fernando CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." (2001).
[24]Suzuki, Jun, and Hideki Isozaki. "Semi-supervised sequential labeling and segmentation using giga-word scale unlabeled data." *Proceedings of ACL-08: HLT* (2008): 665-673.
[25]McCallum, Andrew, and Wei Li. "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons." *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 2003.
[26]Luo, Gang, et al. "Joint entity recognition and disambiguation." *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015.
[27]Leek, Timothy Robert. *Information extraction using hidden Markov models*. MS thesis. University of California, San Diego, 1997.
[28]Borthwick, Andrew, and Ralph Grishman. *A maximum entropy approach to named entity recognition*. Diss. New York University, Graduate School of Arts and Science, 1999.
[29]Riaz, Kashif. "Rule-based named entity recognition in Urdu." In *Proceedings of the 2010 named entities workshop*, pp. 126-135. Association for Computational Linguistics, 2010.
[30]Chieu, Hai Leong, Hwee Tou Ng, and Yoong Keok Lee. "Closing the gap: Learning-based information extraction rivaling knowledge-engineering methods." *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 200
[31]Collobert, Ronan, and Jason Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning." *Proceedings of the 25th international conference on Machine learning*. ACM, 2008.
[32]Bengio, Yoshua, et al. "Greedy layer-wise training of deep networks." *Advances in neural information processing systems*. 2007.
[33]Finkel, Jenny Rose, Trond Grenager, and Christopher Manning. "Incorporating non-local information into information extraction systems by gibbs sampling." *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2005.
[34]Finkel, Jenny Rose, and Christopher D. Manning. "Joint parsing and named entity recognition." *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2009.
[35]Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013.
[36]Sil, Avirup, and Alexander Yates. "Re-ranking for joint named-entity recognition and linking." *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 2013.
[37]Huang, Zhiheng, Wei Xu, and Kai Yu. "Bidirectional LSTM-CRF models for sequence tagging." *arXiv preprint arXiv:1508.01991* (2015).
[38]Strubell, Emma, et al. "Fast and accurate sequence labeling with iterated dilated convolutions." *arXiv preprint arXiv:1702.02098* (2017).
[39]Turian, J., Ratinov, L., & Bengio, Y. (2010, July). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 384-394). Association for Computational Linguistics.

[40]Dyer, Chris, et al. "Transition-based dependency parsing with stack long short-term memory." *arXiv preprint arXiv:1505.08075* (2015).
[41]Lin, Dekang, and Xiaoyun Wu. "Phrase clustering for discriminative learning." *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, 2009.
[42]Ratnaparkhi, Adwait. "A maximum entropy model for part-of-speech tagging." *Conference on Empirical Methods in Natural Language Processing*. 1996.
[43]Goller, Christoph, and Andreas Kuchler. "Learning task-dependent distributed representations by backpropagation through structure." *Neural Networks, 1996., IEEE International Conference on*. Vol. 1. IEEE, 1996.
[44]LeCun, Yann, et al. "Backpropagation applied to handwritten zip code recognition." *Neural computation* 1.4 (1989): 541-551.
[45]Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." *science*313.5786 (2006): 504-507.