<Title and authors> **SEQUENCE LABELING and SENTENCE ANALYSIS for NLP RELATED USE CASES**

**Samartha Kajoor Venkatramana, Larry Okeke, Siddharth Sharma**

**14th April 2018.**

**Source paper: http://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf**

**Abstract:**

We propose a learning algorithm **no we do not propose an ago. We propose a set of experiments and perform them on a specific dataset and we use glove embeddings to process it.**  that can be applied on natural language processing, which will enable us perform tasks like: part-of-speech tagging, chunking and named entity recognition. The system we trained learns representation based on labelled training data. This work is then used as a basis for building a freely available tagging system with good performance and minimal computational requirements.

**<Add a few more sentences - SAM>**

Given an annotated dataset of named entities, we train, test and validate a few Machine learning algorithms to perform one of the above-mentioned task, NER, POS or chunking, using the available library vsmlib for processing sentences into word vectors and then using these annotations and the word embeddings to train and generate predictions on the dataset or other similar dataset. Next, we will try to use the above said systems to implement basic user command processing or to understand the user’s sentences.

**1 Introduction:**

The object of this project was to extract simpler representations about the restricted aspects of textual information. These representations are motivated by our understanding that they will help us capture important general information about natural language. They help us describe syntactic information (e.g.: part-of-speech tags, chunking and parsing) or semantic information (e.g.: Named entity recognition).

Most of these states of the art systems help us address a benchmark task by applying linear statistical models. The researchers themselves discover intermediate representation by engineering task-specific features. These features are derived from the output of pre-existing systems, leading to complex runtime dependencies. Our desire to avoid task specific engineered features led us to ignore a large body of linguistic knowledge. The approach that we have implied is “Almost from the scratch” to emphasize the reduced reliance on NLP knowledge. To employ our project, we are going to try to excel on multiple ***benchmark*** tasks, and these ***benchmark*** tasks are going to be viewed as indirect measurements of the internal representation discovered by the learning procedure.

**<Mention the flow of the paper - Larry> <refer: to final project guidelines on black board – Larry>**

**Introduction (1 page) -> Background -> Implementation/Approach -> Results**

In this section, we will introduce the standard NLP tasks which we will view as our benchmark tasks.

They are: 1. Part-Of-Speech tagging (POS), chunking (CHUNK), Named Entity Recognition (NER). We will setup experimentation and will overview the system on this setup.

The experiment setup is described in Table 1, while the state-of-the-art systems are described in Table 2.

**1.1 Part of Speech Tagging:**

In POS we label each word with a unique tag that indicate its *syntactical* role, for ex, noun, pronoun, verb, …A standard benchmark setup is described by Toutanova et al. (2003). Section 0-18 of Wall Street Journal (WSJ) data are used for training, while sections 19-21 are for validation and section 22-24 are for testing.

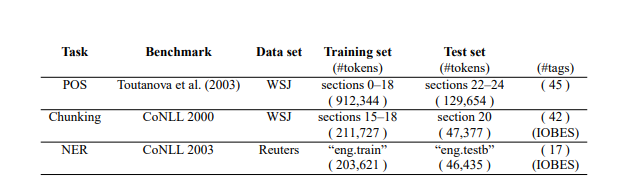


Table1: Experimental setup: for each task, we report the standard benchmark we used, the data set it relates to, as well as training and test information

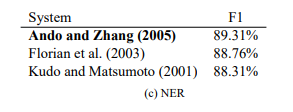
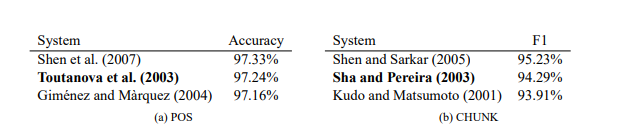


Table2: State of the art systems on three NLP tasks, An F1 Score is calculated for CHUNK and NER and performance is reported with per-word accuracy in the POS.

The POS tags that were used during this project had their classifiers trained on window of text, they are fed to a bidirectional decoding algorithm during inference. The objective is to include the preceding and the following tag. Considering the background, features to deal with unknown words. The reason for using Toutanova et al. (2003) is because it uses maximum entropy classifiers and inference in a bidirectional dependency network to reach a 97.4% accuracy for every word. Shen et al. (2007) published a technique called guided learning, which increased the accuracy up to 97.33% using a technique called guided learning.

**<rough idea of Write what we did here for POS - SAM>**

**1.2: Chunking:**

Chucking is used to label of the sentence with labels such as noun or verb phrases (NP or VP). Every word has only one tag, which is coded as begin-chunk (e.g., B-NP) or inside-chunk tag (e.g., I-NP). Chunking is evaluated using the CoNLL 2000 shared task. WSJ are used for training and testing. We validate by splitting the training set.  
F1-Score is used to understand how well the dataset was used for chunking. SVM’s can be used for chunking and finalising an F1-Score, Kudoh and Matsumoto (2000) got an F1 score of 93.48% using Support Vector Machines (SVMs). In SVM, each machine was trained pair by pair and fed with a word of interest which was POS and words as features and its preceding and following tags.

**F1 = 2 · (precision · recall) / (precision + recall)**

We could also use POS features from an outside tag, or carefully craft features which change the data representation by adding some tags together with their POS representation. Which was used by Shen and Sarkar (2005) which obtained them an F1-score of 95.23%.

**< rough idea of Write what we did here for Chunking -SAM>**

**1.3: Named Entity Recognition:**

NER marks nuclear components in the sentence into classes, for example, "PERSON" or "LOCATION". As in the chunking undertaking, each word is relegated a tag prepended by a pointer of the start or within an element. The CoNLL 2003 setup3 is a NER benchmark informational collection considering Reuters information. The challenge gives training, approval and testing datasets.

Afterward, Ando and Zhang (2005) achieved 89.31% F1 with a semi-managed approach. They prepared together a direct model on NER with a straight model on two assistant unsupervised undertakings. They likewise performed Viterbi decoding at test time. The unlabelled collection was 27M words taken from Reuters. Highlights included words, POS labels, postfixes and prefixes or CHUNK labels, however in general were less particular than CoNLL 2003 challengers.

**< rough idea of Write how we are using NER -SAM>**

**2:** **Background:**

**<1/2 page>  
 2.1: CoNLL 2003:**

CoNLL is a shared task entity: which is language-independent named recognition. Named elements are phrases that contain the names of people, associations and areas.

For Instance, let us consider the example:

**[ORG U.N.] official [PER Ekeus] sets out toward [LOC Baghdad].**

This sentence contains three named elements: Ekeus is a person, U.N. is an organization and Baghdad is a location. Named substance acknowledgment is a critical errand of data extraction frameworks. There has been a considerable measure of work on named element acknowledgment, particularly for English (see Borthwick (1999) for an outline). The Message Understanding Meetings (MUC) have offered designers the chance to assess frameworks for English on similar information in an opposition. They have additionally delivered a plan for element comment (Chinchor et al., 1999). More as of late, there have been other framework advancement rivalries which managed distinctive dialects (IREX and CoNLL-2002).

 The CoNLL-2003 named information comprises of eight records covering two dialects: English and German1. For our project we will be using the English dialect. For every one of the dialects there is a preparation document, an advancement record, a test record and an extensive record with unannotated information. The learning strategies were trained with the training data. The improvement information could be utilized for tuning the parameters of the learning strategies. The test of the current year's shared task was to consolidate the unannotated information in the learning procedure somehow. At the point when the best parameters were discovered, the strategy could be prepared on the preparation information and tried on the test information. The consequences of the distinctive learning strategies on the test sets are thought about in the assessment of the mutual errand. The split between advancement information and test information was kept away from frameworks being tuned to the test information.

**2.2: Word Embeddings:**

Word Embeddings is the aggregate name for an arrangement of dialect modelling and highlight learning systems in characteristic dialect training in Natural Language Processing where words or expressions from the vocabulary are mapped to vectors of genuine numbers. Reasonably it includes a numerical implanting from a space with one measurement for every word to a consistent vector space with considerably higher measurement. Strategies to create this mapping incorporate neural networks, dimensionality lessening on the word co-event matrix, probabilistic models, and unequivocal portrayal as far as the setting in which words appear. Word and expression embeddings, when utilized as the fundamental information portrayal, have been appeared to support the execution in NLP errands, for example, syntactic parsing and sentiment analysis.

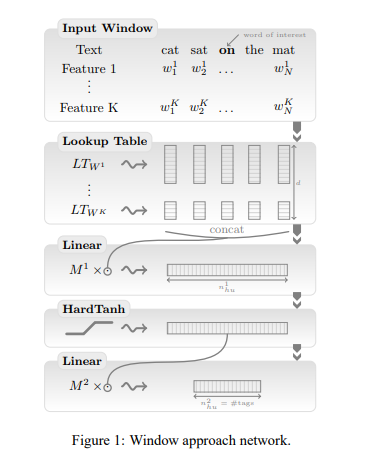
**3. Approach:**

In our examinations, we focused on the standard assessment method of each CoNLL challenges for NER, CHUNK and POS. Specifically, we picked the hyper-parameters of our model as per a basic approval method i.e. Accuracy score for POS and F1-Score for NER and CHUNK. All these three undertakings are assessed by our models. The POS is assessed by registering the per-word exactness, as it is the situation for the standard benchmark we look at (Toutanova et al., 2003). We utilized the CoNLL evaluation for assessing POS, NER and Chunk.

**<2 Pages>**

**< Approach must Include - SAM>**

**<Include this drawing, from page 7 of http://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf >**



[[[Describe your methods. ‣ Include explanations, diagrams, formulas, and any other items that will allow reader to understand your work ✦ Assume reader has general familiarity with machine-learning domain — use citations as needed]]]

**<fill 2 pages: SAM>**

**<Type what you know, and we’ll structure it. Send ASAP>**

**4. Result:**

**<2 pages>**

**NOTE: Outline of the result must include**

[[[Dataset ✦ Specifics of data used, as appropriate ‣ Experiments and performance evaluation ✦ Describe experiments performed ‣ Describe results fully and clearly ✦ Include graphs, plots, and any images/figures to support your results and conclusions ‣ Discussion ✦ Conclusions from your results ✦ Your results and conclusions in broader context (if applicable) ✦ Recommended future directions (as applicable)]]

**<We will all sit together and do it. – Larry, let us know when you’re available>**

**5.** Currently we have a lot of existing systems to perform Chunking, POS and NER. We use a subset of these tools and then utilize the existing libraries in python, which help to perform the required task. Many research papers talk about analysis of active learning strategies for such tasks. Such comparisons help to make the right choice when we want to improve the accuracy and when we need to decide what exactly we can achieve using these methods and tools. We will utilize CONLL dataset to evaluate the performance of word embeddings on sequence labelling tasks. We build the classifier train it and test it and get the performance metrics. Later, we might upgrade the program to perform sentence analysis for helping our system interact and understand the user’s words and commands or we might implement an ANN/RNN to do the same.

**2 Related Work:**

Images about the stuff we’re gonna do. Examples of things done by previous researchers in this field.

**3 Approach:**

**3.1: breaking down approach.**

**3.2: Breaking down part 2.**

**3.3: Breaking down part 3.**

**4.Model Architecture:**

**5 Training Details:**

**6 Experiments and Results**

**7 Conclusion:**

Summary of what/why/how and main conclusions of your project ‣ Concise “take-away message”

**<1/2 Page -Larry>**

**6. 8 Future work**

**9 Acknowledgements:**

**References:**

<Siddharth>

Tjong Kim Sang, E. F., & Buchholz, S. (2000, September). Introduction to the CoNLL-2000 shared task: Chunking. In Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7 (pp. 127-132). Association for Computational Linguistics.