Appendix A: Graph Based Semi-Supervised Learning for Tamil POS Tagging

Mokanarangan Thayaparan, Surangika Ranathunga, Uthayasanker Thayasivam
Dept. of Computer Science and Engineering
University of Moratuwa, Katubedda 10400, Sri Lanka
{mokanarangan.11, surangika, rtuthaya }@cse.mrt.ac.lk

Abstract

Parts of Speech (POS) tagging is an important pre-requisite for various Natural Language Processing tasks. POS tagging is rather challenging for morphologically rich languages such as Tamil. Being low-resourced, Tamil does not have a large POS annotated corpus to build good quality POS taggers using supervised machine learning techniques. In order to gain the maximum out of the existing Tamil POS tagged corpora, we have developed a graph-based semi-supervised learning approach to classify unlabelled data by exploiting a small sized POS labelled data set. In this approach, both labelled and unlabelled data are converted to vectors using word embeddings and a weighted graph is constructed using Mahalanobis distance. Then semi-supervised learning (SSL) algorithms are used to classify the unlabelled data. We were able to gain an accuracy of 0.8743 over an accuracy of 0.7333 produced by a CRF tagger for the same limited size corpus.

Keywords: Semi-Supervised Learning, Low-resourced languages, Graph-based SSL, Word Embedding, POS tagging

1. Introduction

In the recent past, supervised learning methods have produced high accuracies for Parts-of-Speech (POS) tagging (Gimenez and Marquez, 2004). In particular, sequence models such as hidden Markov models (HMM) and conditional random fields (CRF) have given good results (Huang et al., 2015). However, these techniques rely on the availability of relatively large amounts of annotated data. Hence, building an accurate domain insensitive POS tagger is challenging for low resourced languages.

Tamil is one such low resourced language, which is widely used in South India and Sri Lanka. There have been several POS taggers developed for Tamil language using supervised learning techniques (Dhanalakshmi et al., 2009)(Pandian and Geetha, 2009). Since the annotated corpora used in this research have been of small size and from a single domain, these supervised techniques greatly suffer from accuracy and domain adaptability (Rani et al., 2016). For example, FIRE corpus (Forum for Information Retrieval Evaluation, 2014), a widely used freely available Tamil POS annotated corpus contains only 80k words. In contrast, the Wall Street corpus, which is an English POS-annotated corpus has a word count of 1,173K words (Gimenez and Marquez, 2004), meaning that the size of the FIRE corpus is approximately 15 times smaller than the Wall Street corpus. Thus, when using a small corpus such as FIRE, we cannot expect similar accuracy to that of English when supervised techniques are used. Moreover, these approaches depend on language dependent features such as morphological tags (Dhanalakshmi et al., 2009) thus limiting the scalability for adapting to other low resourced languages.

In contrast to supervised approaches, semi-supervised approaches such as graph based semi-supervised learning and manifold regularization (Niyogi, 2013) use both labeled and unlabelled data for their classification, and have proven to work with a small data sets (Zhu et al., 2003). Despite having smaller sized POS-tagged data for Tamil, there has been only two research leveraging the opportunity presented by semi-supervised learning. Ganesh et al. (2014) have used segmentation patterns to implement a bootstrapping approach for POS tagging. This approach relies on language dependent data such as suffix context patterns. Rani et al. (2016) use small annotated training data to build a classifier model using context-based association rule mining. This approach neither includes any language-specific linguistic information nor requires a large corpus. However, they collect all possible words occurring in the same context from the untagged data into a list called context-based list, thus limiting it from scaling to large monolingual corpus.

Graph based semi-supervised learning (SSL) has gained traction in Natural Language Processing tasks such as question answering (Celikyilmaz et al., 2009), structural tagging (Subramanya et al., 2010), and speech language recognition (Liu et al., 2016). Graph based SSL builds a meaningful graph using labelled and unlabelled instances. It then employs an SSL algorithm such as harmonic functions (Zhu et al., 2005) or label propagation (Zhu et al., 2003) to label the unlabelled instances. Graph based SSL is easily parallelizable and scalable to large data (Zhu et al., 2005).

In this paper, we present a novel graph-based semi-supervised approach to produce an accurate POS tagger for Tamil using a limited size corpus. Our idea is inspired by Talukdar and Pereira (2010)'s case study on modified absorption, which is a label propagation algorithm. They have implemented a Named Entity recognizer by building a connected word graph. Similarity between words is measured using WordNet. Then they employ label propagation to assign labels to all the unlabelled nodes.

Since Tamil is a low resourced language with no proper WordNet, we built a connected word graph using word vectors and employed label propagation. Our method is based on the clustering hypothesis that relative distance of word vectors of similar categories is lower than those between different categories. We use neural word embedding (Word2Vec (Mikolov et al., 2013), FastText (Joulin et al., 2016)) to create word vectors. Mahalanobis distance is used for measuring the distance (metric learning) between these vectors in order to construct the graph. Mahalanobis distance generalizes the standard Euclidean distance, and has proven to be more effective (Davis et al., 2007). We empirically tested with four different metric learning algorithms (Information Theoretic Metric Learning (ITML) (Davis et al., 2007), Sparse Determinant Metric Learning (SDML) (Qi et al., 2009), Least Squares Metric Learning (LSML) (Liu et al., 2012), and Local Fisher Discriminant Analysis (LFDA) (Sugiyama, 2006)) to calculate Mahalanobis distance. Once the graph is constructed with labeled and unlabeled nodes, to assign labels to unlabeled nodes, we experimented with three different SSL algorithms (LP-ZGL) (Zhu et al., 2003), Absorption (Talukdar et al., 2008) and Modified Absorption (MAD) (Talukdar and Pereira, 2010)). Local Fisher Discriminant Analysis (LFDA) metric learning coupled with Label Propagation(LP-ZGL) yielded a maximum accuracy of 0.8743 for the FIRE corpus against a baseline accuracy of 0.7338 achieved by using a traditional CRF model. Unlike supervised learning approaches, our approach does not require a large high quality annotated data set, or language dependent features.

Thus the contributions of this paper are: (1) converting words to vectors using neural word embedding and building meaningful word graphs, (2) using Mahalanobis distance to measure relationships between word vectors, hence measuring the correlation between variables, and (3) using a language independent graph based semi-supervised approach for POS tagging in Tamil.

The rest of the paper is organized as follows. Section 2 discusses graph based semi supervised learning techniques and previous attempts on Tamil POS tagging. Section 3 details the data set used in our experiment. Section 4 discusses the methodology and how we implemented the system. Section 5 details the experiments carried out and the relevant results. Section 6 and Section 7 document the conclusion and future work, respectively.

2. Related Work

2.1. Graph based Semi-supervised Learning

Graph theory and Natural Language Processing are well studied disciplines, but are commonly perceived as distinct with different algorithms and with different applications. But recent research has shown that these disciplines are connected and graph-theoretical approaches can be employed to find efficient solutions for NLP problems. Entities are connected by a range of relations in many NLP problems and graph is a natural way to capture the relationship between the entities. Graph based approaches have been used in word sense disambiguation, entity disambiguation, thesaurus construction, textual entailment and semantic classification (Mihalcea and Radev, 2011).

Graph based semi-supervised learning builds graphs connecting labeled and unlabeled data points, and perform classification by propagating the labels. The graph is constructed to reflect our prior knowledge about the domain. The intuition is that similar data points have similar labels. We let the hidden/observed labels be random variables on the nodes of this graph. Labels are injected to unlabeled nodes from labeled nodes. Graphs provide a uniform representation for heterogeneous data and are easily parallelizable (Zhu et al., 2005).

One of the challenges of graph based approach is building the graph that reflects the relationship between entities. Depending on the task, the nodes and edges may represent a variety of language related units and links. Different NLP tasks have approached this challenge in different ways. For the task of opinion summarization, Zhu et al. (2013) constructed a graph of sentences linked by edges whose weight combines the term similarity and objective orientation similarity. And to perform discourse analysis in chat, Elsner and Charniak (2010) predicted the probabilities for pair of utterance as belonging the same conversation thread or not based on lexical, timing and discourse-based features. Then constructed a graph with each nodes representing the utterances and the edges representing the probability score between the nodes. Although these approaches are evidences for the versatility of graph based approaches, these cannot be adopted to a word level problem like sequential tagging. Using graph methods for sequential tagging relies on the belief that similar words will have the same tag. Unlike the aforementioned approaches, here the nodes in these graph represents words or phrases and the the edges will indicate the similarity between nodes. Talukdar and Pereira (2010) tag words with NER information through a label propagation algorithm on a word similarity graph built using WordNet information. Words are represented are the graph vertices and the edge denotes the WordNet relationship. This approach cannot be adopted for a low resource language which doesn't have a proper WordNet. Subramanya et al. (2010) POS tags on a similarity graph where local sequence contexts (n-grams) are vertices. The similarity function between graphs is the cosine distance between the point-wise mutual information vectors (PMI) representing each node. The point-wise mutual information is calculated between n-gram and set of context features. These context features includes suffixes, left word and right word contexts. The challenge of this approach is the scalability for a morphologically complex language like Tamil.

2.2. Tamil POS tagging

Tamil is a low resourced, morphologically rich language with many inflections and a complex grammatical structure. Thus, automatic POS tagging for Tamil is a challenging task. Supervised learning approaches have been heavily undertaken in Tamil for POS tagging. These include CRF models using morphological information (Pandian and Geetha, 2009) and Support Vector Machines (SVM) using semantic information (Dhanalakshmi et al., 2009). These models had been trained using different corpora containing approximately 200k annotated words. These annotated corpora or taggers are not publicly available.

There have been very few attempts in using semi-supervised approaches for Tamil language to develop POS taggers. Ganesh et al. (2014) have used language features with a bootstrapping approach to obtain a precision of 86.74%. They have presented a pattern based bootstrapping approach using only a small set of POS labelled suffix context patterns. The patterns consist of a stem and a sequence of suffixes, obtained by segmentation using a manually created suffix list. This bootstrapping technique generates new patterns by iteratively masking suffixes with low probability of occurrences in the suffix context, and replacing them with other co-occurring suffixes. This approach relies on language specific information.

Rani et al. (2016) have employed a semi-supervised rule mining approach using morphological features for Hindi, Tamil, and Telugu languages. They have used a combination of a small annotated and untagged training data to build a classifier model using a concept of context-based association rule mining. These association rules work as context-based tagging rules.

3. Data set

For our experiment, we used the FIRE Tamil Corpus. The FIRE Tamil corpus contains 80k POS tagged words with 21 different tags as shown in Table 1.

NN	Noun
NNC	Compound Noun
RB	Adverb
VM	Verb Main
SYM	Symbol
PRP	Personal Pronoun
JJ	Adjective
NNP	Pronoun
PSP	Prepositions
QC	Quantity Count
VAUX	Verb Auxiliary
DEM	Determiners
QF	Quantifiers
NEG	Negatives
QO	Quantity Order
WQ	Word Question
INTF	Intensifier
NNPC	Compound Pro Noun
CC	Coordinating Conjunction
RBP	Adverb Phrase

Table 1: POS tagsets for FIRE Tamil Corpus

4. Methodology

Our work is inspired by Talukdar and Pereira (2010)'s case study on the performance of different algorithms for classification in graphs. In this work, words are represented as nodes and the similarity between nodes are measured using WordNet distance. Since Tamil is a low resourced language, this approach was not viable for us. Another approach was to represent words by converting them to vectors and computing the similarity. Subramanya et al. (2010) had employed a point wise mutual information (PMI) based approach to convert the word to vectors and compute the similarity by measuring the cosine distance. His approach used hand-crafted features that will not work with same efficiency across different languages.

Hence, an efficient way of representing a word in the vector space has to be determined. In addition, it is required to identify mechanisms for (1) constructing a meaningful graph based on the word vector, and (2) classifying unlabelled words based on the constructed graph by measuring the similarity.

4.1. Representing a word in the vector space

We adopted the Word2Vec model proposed by Mikolov et al. (2013) and convert the word into the vector space to construct the graph. To the best of our knowledge, Word2Vec has never been used to construct weighted word graphs to be used in SSL. Similarly we also experimented with Fast Text skipgram (Bojanowski et al., 2016) and bag of words models (Joulin et al., 2016). The key difference between Word2Vec and FastText is that Word2Vec treats each word in corpus as an atomic entity and generates a vector for each word. In contrast, FastText treats each word as composed of ngrams and the vector word is made of the sum of these vectors.

4.2. Constructing a meaningful graph based on the word vector

Each word is converted to a d dimensional vector space. Out of the n words in the list, n_l are labelled($n >>> n_l$). We employ 32 different tags to denote each POS entity (Dhanalakshmi et al., 2009). $G = (V, E, W)$ is the graph we are interested in constructing; where V is the set of vertices with $|V| = n$, E is the set of edges. W is the symmetric $n \times n$ matrix of edge weights we want to learn. Usually we could choose a standard distance metric (Euclidean, City-Block, Cosine, etc.). Instead, Mahalanobis distance has proven to be effective with clustering problems over the standard metrics (De Maesschalck et al., 2000).

We use a supervised method for learning the Mahalanobis distance. For this purpose, we need to calculate the positive definite matrix A of size $d \times n$ that parametrizes the Mahalanobis distance, $d_A(x_i, x_j)$ (Dhillon et al., 2010; Davis et al., 2007; Sugiyama, 2006) between words x_i and x_j as shown in Equation (1).

$$d_A(x_i, x_j) = (x_i - xj)^T A(x_i - x_j) \quad (1)$$

Since A is positive definite, it can be decomposed into $P^T P$, where P is another matrix of size $d \times d$

$$\begin{aligned} d_A(x_i, x_j) &= (x_i - x_j)^T P^T P(x_i - x_j) \\ &= (Px_i - Px_j)^T (Px_i - Px_j) \\ &= d_I(Px_i, Px_j) \end{aligned} \quad (2)$$

There are many proposed methods for calculating the transformation matrix P. We empirically experimented with different metric learning algorithms, including Information Theoretic Metric Learning (ITML) (Davis et al., 2007), Sparse Determinant Metric Learning (SDML) (Qi et al., 2009), Least Squares Metric Learning (LSML) (Liu et al., 2012), and Local Fisher Discriminant Analysis (LFDA) (Sugiyama, 2006).Researches in link prediction in networks (Shaw et al., 2011), music recommendation (McFee et al., 2011) and bio metrics verification (Ben et al., 2012) has shown that metric learning plays a vital role increasing accuracy of the system.

ITML minimizes the differential entropy between multivariate Gaussian under constraints on the distance function. Davis et al. (2007) have expressed the problem as that of minimizing the LogDet divergence subject to linear constraints. SDML uses l_1-penalized log-determinant regularization to calculate the metric. This algorithm exploits the sparsity nature underlying the intrinsic high dimensional feature space. LSML uses an algorithm that minimizes a convex objective function corresponding to the sum of squared residuals of constraints. Finally LFDA, is a linear supervised dimensionality reduction method which is particularly useful when dealing with cases where one or more core classes consist of separate clusters in input space.

We calculate P using each of these metric learning algorithms and project the words into a new space to calculate Px_i. Based on Equation 2, we compute the Euclidean distance in the linearly transformed matrix. Gaussian kernel $[2, 16]$ was used to compute the similarity between words as shown in Equation 3 (Dhillon et al., 2010). We then sparsify the graph by selecting k neighbors for each node and set the weights to zero for all others (Zhu et al., 2003).

$$W_{ij} = exp(\frac{-d_A(x_i, x_j)}{2\sigma^2}) \tag{3}$$

The culmination of all these steps results in a meaningful graph where relative distances of word vectors of similar categories will be lower than those between different categories.

4.3. Classifying Unlabelled Nodes based on the Constructed Graph

Once the graph is constructed, unlabelled words in the graph should be classified. For this, we experimented with Label Propagation(LP-ZGL), and Absorption and Modified Absorption (MAD) techniques. LP-ZGL (Zhu et al., 2003) was one of the first graph based SSL methods. LP-ZGL propagates the labels over the graph by penalizing any label assignment where two nodes connected by a highly weighted edge are assigned different labels. LP-ZGL prefers smooth labeling over the graph. This property is also shared by the other two algorithms. Absorption (Talukdar et al., 2008) has been used for open domain class-instance acquisition. Absorption is an iterative algorithm where label estimates depend on the previous iteration. Modified Absorption (MAD) (Talukdar and Pereira, 2010) shares the same properties of the Absorption algorithm but can be expressed as an unconstrained optimization problem. We experimented with all these algorithms to estimate the labels of the untagged words.

5. Experiments and Results

5.1. Experiments

We split the data into 60k words for training and 20k words for testing. To the best of our knowledge, there has been only Named Entity Recognition research (Abinaya et al., 2014) done in Tamil using FIRE corpus and no POS tagging research done.

We trained both Word2Vec and FastText models with a word window of three (the commonly used window size) using the Tamil Wikipedia corpus (Wikipedia, 2016) (about 1M words) after removing only the punctuation marks. We used these models to convert word to vector form. Each vector is of 300 dimensions. For graph construction, a subset of 3000 sentences with approximately 50k unlabelled words from the Tamil Wikipedia corpus were added to the set. We constructed the word graphs using the aforementioned four metric learning approaches and employed three labeled propagation approaches to identify the best combination.

Since most of the successful approaches related to Tamil POS tagging have been carried out using Conditional Random Fields (CRF) (Pandian and Geetha, 2009), we used the same approach with word trigram feature as our baseline method. Here, trigrams were selected because for Word2Vec and FastText models also, a word window of three was used.

5.2. Results

The following Tables 2-5 document the results obtained for each graph construction algorithm in combination with the classification methods.

Word To Vector Algorithm	MAD	Abs	LP-ZGL
Word2Vec (SkipGram)	0.7534	0.7531	0.7201
Word2Vec (Bag of words)	0.6945	0.6967	0.6754
Fasttext (SkipGram)	0.8146	0.814	0.822
Fasttext (Bag of Words)	0.795	0.7952	0.801

Table 2: Accuracy of Information Theoretic Metric Learning

Word To Vector Algorithm	MAD	Abs	LP-ZGL
Word2Vec (SkipGram)	0.7012	0.701	0.721
Word2Vec (Bag of words)	0.6641	0.6542	0.665
Fasttext (SkipGram)	0.7886	0.7935	0.7988
Fasttext (Bag of Words)	0.7712	0.775	0.7767

Table 3: Accuracy of Sparse Determinant Metric Learning

Word To Vector Algorithm	MAD	Abs	LP-ZGL
Word2Vec (SkipGram)	0.734	0.733	0.732
Word2Vec (Bag of words)	0.701	0.71	0.711
Fasttext (SkipGram)	0.8547	0.861	0.8634
Fasttext (Bag of Words)	0.823	0.834	0.845

Table 4: Accuracy of Least Squares Metric Learning

Word To Vector Algorithm	MAD	Abs	LP-ZGL
Word2Vec (SkipGram)	0.7678	0.7775	0.7757
Word2Vec (Bag of words)	0.7664	0.7567	0.7456
Fasttext (SkipGram)	0.8673	0.8573	**0.8743**
Fasttext (Bag of Words)	0.85	0.853	0.86

Table 5: Accuracy of Local Fisher Discriminant Analysis

As illustrated above, Local Fisher Discriminant Analysis(LFDA) combined with Label propagation yields the best accuracy of 0.8743. LFDA is a linear supervised dimensionality reduction method. It proved effective in our case since each of our words had a size of 300 dimensions. FastText(skipgram) in combination with label propagation consistently performed better than other algorithms in all graph construction methodologies.
To test the robustness of the approach, we trained the best performing combination (LFDA and LP-ZGL) with 20k words and tested with 60k words. It yielded an accuracy of 0.753. Meanwhile, the baseline CRF model only gave an accuracy score of 0.633. This proves that our approach is more robust even when the labelled data set is comparatively small.

6. Conclusion

Our research establishes the fact that graph based semi-supervised approaches are more robust than supervised classification algorithms for POS tagging when the data set is relatively small. Thus graph based semi supervised data can be employed in the early stages of creating POS tagged data sets. Human annotators can correct the automatically annotated corpus with less effort, and the corrected annotated data set can be used in an iterative manner to re-train the tagger. Thus, graph based semi-supervised approaches are particularly useful for POS tagging of low-resourced languages such as Tamil. We used neural word embedding to create a vector representation of words, and Mahanalobis distance to measure distance between word vectors in order to build the graph. This shows that word embedding provides an excellent alternative for WordNet in measuring similarity between words, especially for languages that do not have a WordNet. This is useful not only for graph building, but for any task that requires measuring the similarity of words.

7. Future work

Our language independent work has shown promise with low resources. We have only done the research for one language, and this research should be extended to other languages to verify the general applicability of the presented methodology. We hope to extend this idea for other low resourced sequential tagging problems such as Named Entity Recognition. This research can also be extended to improve and incorporate other word embedding techniques such as VarEmbed that uses morphological priors for probabilistic neural word embedding (Bhatia et al., 2016). We can also experiment with other graph construction algorithms such as b-matching (Jebara et al., 2009). The main limitation of this technique is the amount of time taken to build the graph. Thus we intend to look into different code optimization methods. While we have compared our approach with the pure CRF implementation, Lample et al. (2016) has shown that CRF in combination with LSTM can provide a higher accuracy for Named entity recognition but that approach has not been tried for POS tagging in morphologically complex languages such as Tamil. We are eager to see how our approach stacks up with them.

8. Acknowledgement

This research is partially funded by the National Languages Processing Centre of University of Moratuwa, Sri Lanka. The authors would also like to thank the LKDomain registry for partially funding this publication.

9. Bibliographical References

Abinaya, N., John, N., Ganesh, B. H., Kumar, A. M., and Soman, K. (2014). Amrita_cen@ fire-2014: Named entity recognition for indian languages using rich features. In *Proceedings of the Forum for Information Retrieval Evaluation*, pages 103–111. ACM.

Ben, X., Meng, W., Yan, R., and Wang, K. (2012). An improved biometrics technique based on metric learning approach. *Neurocomputing*, 97:44 – 51.

Bhatia, P., Guthrie, R., and Eisenstein, J. (2016). Morphological priors for probabilistic neural word embeddings. 3 August.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Celikyilmaz, A., Thint, M., and Huang, Z. (2009). A graph-based semi-supervised learning for question-answering. In *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*, ACL '09, pages 719–727, Stroudsburg, PA, USA. Association for Computational Linguistics.

Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. (2007). Information-theoretic metric learning. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 209–216, New York, NY, USA. ACM.

De Maesschalck, R., Jouan-Rimbaud, D., and Massart, D. L. (2000). The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18.

Dhanalakshmi, V., Rajendran, S., Soman, K. P., and Edu, K. (2009). POS tagger and chunker for tamil language.

Dhillon, P. S., Talukdar, P. P., and Crammer, K. (2010). Inference driven metric learning (idml) for graph construction.

Elsner, M. and Charniak, E. (2010). Disentangling chat. *Comput. Linguist.*, 36(3):389–409, September.

Ganesh, J., Parthasarathi, R., Geetha, T. V., and Balaji, J. (2014). Pattern based bootstrapping technique for tamil POS tagging. In *Mining Intelligence and Knowledge Exploration*, Lecture Notes in Computer Science, pages 256–267. Springer, Cham.

Gimenez, J. and Marquez, L. (2004). Svmtool: A general pos tagger generator based on support vector machines. In *In Proceedings of the 4th International Conference on Language Resources and Evaluation*.

Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Jebara, T., Wang, J., and Chang, S.-F. (2009). Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 441–448, New York, NY, USA. ACM.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June. Association for Computational Linguistics.

Liu, E. Y., Guo, Z., Zhang, X., Jojic, V., and Wang, W. (2012). Metric learning from relative comparisons by minimizing squared residual. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, ICDM '12, pages 978–983, Washington, DC, USA. IEEE Computer Society.

Liu, Y., Kirchhoff, K., Liu, Y., and Kirchhoff, K. (2016). Graph-based semisupervised learning for acoustic modeling in automatic speech recognition. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 24(11):1946–1956, November.

McFee, B., Barrington, L., and Lanckriet, G. R. G. (2011). Learning content similarity for music recommendation. *CoRR*, abs/1105.2344.

Mihalcea, R. F. and Radev, D. R. (2011). *Graph-based Natural Language Processing and Information Retrieval*. Cambridge University Press, New York, NY, USA, 1st edition.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. 16 January.

Niyogi, P. (2013). Manifold regularization and semi-supervised learning: Some theoretical analyses. *Journal of Machine Learning Research*, 14:1229–1250.

Pandian, S. L. and Geetha, T. V. (2009). Crf models for tamil part of speech tagging and chunking. In *Proceedinghs of the 22Nd International Conference on Computer Processing of Oriental Languages. Language Technology for the Knowledge-based Economy*, ICCPOL '09, pages 11–22, Berlin, Heidelberg. Springer-Verlag.

Qi, G.-J., Tang, J., Zha, Z.-J., Chua, T.-S., and Zhang, H.-J. (2009). An efficient sparse metric learning in high-dimensional space via l1-penalized log-determinant regularization. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 841–848, New York, NY, USA. ACM.

Rani, P., Pudi, V., and Sharma, D. M. (2016). A semi-supervised associative classification method for POS tagging. *Int J Data Sci Anal*, 1(2):123–136, 1 July.

Shaw, B., Huang, B., and Jebara, T. (2011). Learning a distance metric from a network. In J. Shawe-Taylor, et al., editors, *Advances in Neural Information Processing Systems 24*, pages 1899–1907. Curran Associates, Inc.

Subramanya, A., Petrov, S., and Pereira, F. (2010). Efficient graph-based semi-supervised learning of structured tagging models. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 167–176, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sugiyama, M. (2006). Local fisher discriminant analysis for supervised dimensionality reduction. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 905–912, New York, NY, USA. ACM.

Talukdar, P. P. and Pereira, F. (2010). Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1473–1481, Stroudsburg, PA, USA. Association for Computational Linguistics.

Talukdar, P. P., Reisinger, J., Paşca, M., Ravichandran, D., Bhagat, R., and Pereira, F. (2008). Weakly-supervised acquisition of labeled class instances using graph random walks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 582–590, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhu, X., Ghahramani, Z., and Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pages 912–919. AAAI Press.

Zhu, X., Lafferty, J., and Rosenfeld, R. (2005). *Semi-supervised learning with graphs*. Ph.D. thesis, Carnegie Mellon University, language technologies institute, school of computer science.

Zhu, L., Gao, S., Pan, S. J., Li, H., Deng, D., and Shahabi, C. (2013). Graph-based informative-sentence selection for opinion summarization. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '13, pages 408–412, New York, NY, USA. ACM.

10. Language Resource References

Forum for Information Retrieval Evaluation. (2014). *FIRE Corpus*. Indian Institute of Science, Bangalore.

Wikipedia. (2016). *Tamil Wikipedia Corpus*. Wikipedia.

Appendix B: Configuration File Used to Build CRF Tagger with AllenNLP

```
{
    "dataset_reader":{
        "type":"sequence_tagging",
        "word_tag_delimiter":"\t",
        "token_delimiter":"\n",
        "token_indexers":{
            "tokens":{
                "type":"single_id"
            },
            "elmo":{
                "type":"elmo_characters"
            },
            "token_characters":{
                "type":"characters"
            }
        }
    },
    "train_data_path":"/src/data/Tamil_NER_Clean/train.txt
    ",
    "validation_data_path":"/src/data/Tamil_NER_Clean/dev.
    txt",
    "test_data_path":"/src/data/Tamil_NER_Clean/test.txt",
    "evaluate_on_test":true,
    "model":{
        "type":"crf_tagger",
        "text_field_embedder":{
```

```
        "tokens":{
            "type":"embedding",
            "embedding_dim":300,
            "pretrained_file":"/src/vectors/Wang_Tamil.
txt.gz"
       },
        "elmo":{
            "type":"elmo_token_embedder",
            "options_file":"/src/options.json",
            "weight_file":"/src/vectors/tamil_elmo.hdf5",
            "do_layer_norm":false,
            "dropout":0.5
        },
        "token_characters":{
            "type":"character_encoding",
            "embedding":{
                "embedding_dim":25
            },
            "encoder":{
                "type":"gru",
                "input_size":25,
                "hidden_size":80,
                "num_layers":2,
                "dropout":0.25,
                "bidirectional":true
            }
        }
    },
    "encoder":{
        "type":"gru",
```

```
            "input_size":1484,
            "hidden_size":300,
            "num_layers":2,
            "dropout":0.25,
            "bidirectional":true
        },
        "regularizer":[
            [
                "transitions",
                {
                    "type":"l2",
                    "alpha":0.01
                }
            ]
        ]
    },
    "iterator":{
        "type":"basic",
        "batch_size":32
    },
    "trainer":{
        "optimizer":"adam",
        "num_epochs":20,
        "patience":10,
        "cuda_device":-1
    }
}
```

Listing 1: CRF tagger configuration

Appendix C: Graph based semi-supervised sequence tagging for low resourced languages

Anonymous EMNLP submission

Abstract

We present a novel Graph-based Semi-Supervised Learning (GSSL) approach for sequence tagging tasks. Performance gains over traditional GSSL techniques are achieved by capturing local context information in graph representation, as well as by producing a low-dimensional graph representation that separates nodes belonging to distinct categories. This GSSL approach far outperforms the other state-of-the-art techniques in low-resourced settings, thus proving to a viable solution for sequence tagging for low-resourced languages.

1 Introduction

When supervised data is scarce, it has been common to employ semi-supervised learning (SSL) techniques for many different Natural Language Processing (NLP) tasks (Garrette et al., 2013; Cheng et al., 2016). In general, graph-based semi-supervised learning (GSSL) techniques have shown even better performance than other SSL techniques (Subramanya and Bilmes, 2008). Graphs of words capture term dependence, encode the strength of the dependence as edge weights, and capture term order (via directed edges) (Rousseau and Vazirgiannis, 2013; Skianis et al., 2016; Rousseau et al., 2015). Hence, GSSL shows greater potential for NLP tasks. They have been used in word sense disambiguation, entity disambiguation, thesaurus construction, textual entailment, and semantic classification (Mihalcea and Radev, 2011), which suggests that semantic relationships between words have been exploited in graph construction.

As for sequence tagging, there are two key factors in constructing a meaningful graph. First, it is important to be able to represent each word occurrence (token) as a vertex because the label assignment for the same word type may differ based on the context it is used. Second it is important to link vertices that are likely to have the same label, where edge weights govern how strongly the labels of the nodes linked by the edge should agree. Given such a graph, a label propagation algorithm could label the unlabeled vertices based on the information of their nearest neighbours.

Related literature suggests that *types* have been the common choice for representing vertices in the graph (Mihalcea and Tarau, 2004). Early work on using GSSL for sequence tagging problems also relied on this word-based representation (Talukdar and Pereira, 2010), thus missing out context information in their vertex representation. These approaches mostly rely on word based similarity measures to determine edge weights.

The alternative way to represent vertices is using local sequence contexts (n-gram). A notable work along this line was reported by Subramanya et al. (2010), which exploited the empirical observation that the Parts of Speech (POS) of a word occurrence is mostly determined by its local context. They represent each vertex using a vector of pointwise mutual information (PMI) values, computed using the n-gram and each of the features that occur with tokens of that n-gram. The cosine distance between these PMI vectors of a pair of vertices is used as edge weights between those vertices.

Instead of these PMI-based count models, much recent GSSL work for sequence tagging reported the use of traditional neural word embeddings such as WORD2VEC (predict models) for representing vertices of the graph (Mokanarangan et al., 2018; Demirel, 2017). These predict models are much more concise than PMI vectors. However, these traditional WORD2VEC approaches are less sensitive to word order (the local context of a word occurrence), which makes them sub-optimal for sequential learning problems (Ling et al., 2015).

There is another limitation of these approaches, which is not necessarily limited to GSSL methods that use predict models, but applicable for any GSSL method. The foundation assumption in GSSL is that the similar nodes will carry same labels. Even though this assumption is effective in many cases, this is not completely true for many sequence labeling problem instances. For example, the word *'amazed'* and the word *'fantastic'* are semantically very similar but they should be labeled with different POS tags.

We present a novel graph building approach to tackle the above limitation of count models used in GSSL techniques for sequential tagging. We adopt the graph building methodology mentioned in Mokanarangan et al. (2018), but leverage the structured embedding models presented by Ling et al. (2015) and Peters et al. (2017), which are more sensitive to word order. We empirically evaluate some compelling choices for aggregating these n-gram token vectors to represent n-grams effectively.

In order to tackle the second limitation, a graph constructed using this n-gram representation is transformed into a lower dimensional vector space in such a way that vertices belonging to different classes are well-separated. This helps to reduce overall computational complexity as well.

We evaluate our approach for three different sequence tasks (POS, Named Entity Recognition (NER), and Chunking) for English using the CoNLL 2003 data set (Tjong Kim Sang and Buchholz, 2000), and for POS for Sinhala and Tamil. For each experiment, we use 1 million unlabeled tokens. We vary the amount of labelled tokens in a step-wise manner until up to 100,000 tokens, to resemble a low-resourced setting. Results show that our solution outperforms the state-of-the-art techniques for sequence tagging when the amount of training data is less than 80,000 tokens.

2 Related Work

Early work on using GSSL for sequence tagging problems relied on word-based graph representations. Talukdar and Pereira (2010) had constructed a word graph using WordNet to perform NER. In this approach, vertices are noted as surface level word forms and each relationship in WordNet is represented as an edge. Although simple and straightforward, this approach fails to capture the syntactic information essential for sequence classification tasks.

In contrast, Subramanya et al. (2010) represent each vertex using a vector of point-wise mutual information (PMI) values, computed using the n-gram and each of the features that occur with tokens of that n-gram. The cosine distance between these PMI vectors of a pair of vertices are used as edge weights between those vertices. These PMI vectors are capable of capturing local context information. However, they note that the vectors used in this approach are sparse and high dimensional.

Extending on Subramanya et al. (2010)'s work, Das and Petrov (2011) designed unsupervised POS taggers for languages that have no labeled training data. They constructed a graph based on the same PMI features introduced by Subramanya et al. (2010), and used graph-based label propagation for cross-lingual knowledge transfer. This solution was based on the observation that despite the language differences, words in different languages share similar relationships in local context.

In their research on graph-based posterior regularization for semi-supervised structured prediction, He et al. (2013) claimed that using Subramanya et al. (2010)'s features to build graphs leads to unrelated trigrams to match. Instead they proposed a different set of features to build PMI based graphs which also suffers from sparsity.

Recently, Demirel (2017) had proposed an approach to solve POS tagging where every word in a corpus is connected into a graph where each node is denoted by a word embedding vector. They capture the word ordering information by connecting each word to next and previous word in the corpus. This graph is then directly fed into a neural network model called graph convolutional network (GCN) for classification.

Exploiting the cluster assumption of word embedding, Mokanarangan et al. (2018) had proposed an approach where each node is represented by a word embedding vector, and edges between nodes are calculated using supervised metric learning. Though this approach has shown promise in low resourced settings, it fails to capture different context information for the same word.

3 Graph Construction and Label Propagation

3.1 Representing Nodes of the Graph

In sequence tagging problems, label of a word is predominantly determined by its context. Thus, syntactic relationships between word tokens play a major role. For example, the word *present* may appear as a noun or a verb, depending on the context. Thus, without referring to the context, the exact POS tag of the word cannot be determined. As an example with respect to Named Entities (NEs), consider the NEs "Central Bank spokesman" and "The Central African Republic". Here, the word 'Central' is used as part of both an Organization and Location (Peters et al., 2017).

As opposed to using lexical units or simple word vector representations to create nodes, we experiment with different types of vector representations.

Related literature presents contradicting arguments with respect to the performance of count models and predict models. Baroni et al. (2014) and Mikolov et al. (2013) claim that predict models such as WORD2VEC and FASTTEXT capture more syntactic and semantic information compared to traditional count based distributional models such as PMI vectors. However, much recently Levy et al. (2015) have claimed that with proper system choices and hyper parameters, traditional count models can yield similar gains. However, in count models, increasing the unlabeled data produces extremely spares vectors that leads to computationally demanding graph building. Thus we experimented with the following predict models that have claimed to capture syntactic information.

WANG2VEC (Ling et al., 2015): WANG2VEC is presented as a model that captures more syntactic-oriented embedding than WORD2VEC. Though this still produces same vector representations for words in different contexts, experiments have shown that vectors produced are syntactically close.

FASTTEXT (Bojanowski et al., 2016): While WORD2VEC treats each word in corpus as an atomic entity and generates a vector for each word, FASTTEXT treats each word as comprised of n-grams and the vector is made of sum of these vectors. Previous research (Mokanarangan et al., 2018) has shown that FASTTEXT performs well when compared with WORD2VEC in GSSL settings.

ELMO (Peters et al., 2017): This semi-supervised bidirectional language model computes an encoding of the context at each position in the sequence. It has been proved that ELMO surpasses the state of the art approaches in capturing semantic and syntactic models. Although rich with information, it is computationally exhaustive to create these vectors. Unlike other word embedding models used, this model produces vectors for a word based on the contextual information of the word.

As mentioned earlier, we base our work on one assumption that words with same local sequence context will have the same sequence tags. In order to capture the local context information in our graph, we experimented with one solution: concatenation of vector n-grams.

3.2 Creating Edges of the Graph

Similar to the approach proposed by Subramanya et al. (2010), once the nodes in the graph are fixed, the edge weights w_{ij} between them between two vector n-grams i and j are defined as shown in Equation 1.

$$w_{ij} = \begin{cases} sim(i,j), & \text{if } i \in K(j) \text{ or } j \in K(i). \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Here $K(i)$ in the set of k-nearest neighbors of vector n-gram i. The similarity function was defined using the Gaussian kernel denoted in Equation 2 (Dhillon et al., 2010). Here $d(x_i, x_j)$ is the euclidean distance between vectors i and j.

$$sim(i,j) = exp(\frac{-d(x_i, x_j)}{2\sigma^2}) \tag{2}$$

Theoretically, there can be an edge between each pair of nodes in the graph. However, one can safely disregard edges that have very low weights, because the relationship between such nodes is very weak. Such weak edges can add noise to label propagation.

The identification of the set of vertices that should be connected to a given vertex can be modelled in the form of k-nearest neighbour problem, where the objective is to determine the set of vertices that have the strongest relationship with the given node (i.e., we determine the set of edges with the highest weight for a given node). Determining the set of edges using k-nn is more effective if the vertices belonging to different classes are well-separated. Thus we transform the vector space into a lower dimension while preserving the separation of classes.

This dimensionality reduction serves another purpose. The performance of nearest neighbor algorithms degrades when the size of the vector increases. Since we used word embedding models result in 300 dimensions. When concatenating vector n-grams, this dimension reaches 900. Thus the dimensionality reduction makes graph construction extremely efficient.

Algorithm 1 presents the graph construction procedure.

Algorithm 1: GSSL using word embedding

Data: Corpus with n number of words where n_l are labeled ($n >>> n_l$)

for *each* w_i *in corpus* **do**

  $vec_i = ConvertWordToVector(w_i)$;

  $v_i = Concatenate(vec_{i-1}, vec_i, vec_{i+1})$;

end

$V_r = BuildVectorList(v)$;

$V_s = SupervisedReduction(V_r)$;

for *each* v_i *in* V **do**

  $e_i = NearestKVectors(v_i, distance =' euclidean')$;

  $w_i = CalculateWeight(e_i)$

end

$E = BuildEdgeMatrix(e)$;

$W = BuildWeightMatrix(w)$;

Build graph $G = (V, E, W)$;

$Predict(G, n)$

3.3 Label Propagation

Label propagation refers to the process of assigning labels to unlabeled nodes using the labelled nodes. The prior assumption of semi-supervised learning is that nearby points and points on the same structure are likely to have the same labels (Zhu et al., 2003). This is a simple and straightforward approach that have been the staple of semi-supervised learning and have yielded encouraging results.

4 Implementation

As mentioned above, high dimensionality of the vectors and the large size of the sample space severely affect the performance of k-nn algorithm. Thus we resorted to approximate nearest neighbor algorithms(ANN). We use Annoy (Bernhardsson, 2018), which has been empirically shown to work better with large data-sets (Aumüller et al., 2017). k was set to an arbitrary value of 20. It should be noted this ANN's accuracy drops when dimensions of the vector is greater than 100. This attribute played an important role in choosing to reduce dimensions.

To achieve a discriminant feature set in a lower dimension, two dimensionality reduction techniques were experimented with Linear discriminant analysis (LDA) and Fisher linear discriminant analysis (LFDA). Both LDA and LFDA are supervised methods that are useful in finding dimensions which aim at separating the clusters (Sugiyama, 2006).

For label propagation, Harmonic Function (HMN) (Zhu et al., 2003) and Local and Global Consistency (LGC) (Zhou et al., 2003) were experimented with. These are two of the well-established label propagation algorithms that have proven their effectiveness in different contexts (Zhu, 2005).

5 Experiments and Results

5.1 Data set

English. We evaluated our approach on CoNLL2003 NER task (Sang and Meulder, 2003) for POS, NER and Chunking task. We emulated a low resource setting for English by using only 20K, 40K, 60K and 100K data as our training setting as opposed to using the full training data.

Tamil. Tamil belongs to the Dravidian language family, which is used in some parts of South Asia. For Tamil we used the dataset from the Forum for Information Retrieval (FIRE) (Majumder et al., 2008). The dataset has nearly 80K labeled data with 32 POS classes.

Sinhala. Sinhala is an Indo Aryan language predominantly used in Sri Lanka. It has evolved from the same language family as Hindi, but being a language limited to an island nation, it has evolved to have its own characteristics. Sinhala is an ideal example of a low-resourced language. For our experiments, we used the University of Moratuwa (UOM) Sinhala POS corpus (Fernando et al., 2016), which currently has 260K tagged tokens labeled using 32 tags.

5.2 Experiment Setup

Experiments are designed to determine the impact of local context information in graph construction for sequence tagging tasks, and the impact of dimensionality reduction on the same. For English, we test the performance of our solution with respect to POS tagging, NER, and Chunking tasks of the CoNLL 2003 dataset. With respect to Tamil and Sinhala, we experiment only with POS tagging, due to the unavailability of data for other tasks.

The current implementation employs the Continuous Bag of Words (skip-gram) model of FASTTEXT (Bojanowski et al., 2016) to generate word embeddings for English, where the vector dimension is 300.

WANG2VEC models are generated using a part of the wiki dump for all the three languages. Dimension of these vectors is also set to 300.

ELMO model (Peters et al., 2017) of 1024 dimensions was reused. ELMO model was not used for Sinhala and Tamil, since we do not have enough computer capacity required to generate the model.

We have experimented with $n = 3$, when generating vector n-grams. For example when $n = 3$, in the example given in Section 3, the word "Central" will be represented by concatenating the word vectors of "The", "Central", "African", thus adding the context information. Thus we end up with a feature vector of 900 dimensions for FASTTEXT and WANG2VEC, and 3072 for ELMO.

For each language, the graph is constructed using 1 million tokens from an unlabeled corpus, and the labeled text size is varied from 20k to 100k in a step-wise manner.

To show that our GSSL solution works in low-resourced settings better than the state-of-the-art reported in the context of high-resourced settings, we compare our results with the work of Peters et al. (2017). We sampled the same amount of training samples from the CoNLL 2003 Shared Task (Sang and Meulder, 2003). For this experiment, according to the discussion by Peters et al. (2017), we used two bidirectional GRUs with 80 hidden units and 25 dimensional character embeddings for the token character encoder. The sequence layer uses two bidirectional GRUs with 300 hidden units each. For regularization, we add 25% dropout to the input of each GRU, but not to the recurrent connections to setup the model. We also embed the ELMO model to represent each word in this bidirectional model and tested it.

5.3 Results

For POS we report the accuracy, while for Chunking and NER we report the official evaluation metric (micro-averaged F1 score).

Both LDA and LFDA showed near equal performance, and so did HMN and LGC. Thus the following results only showcase the experiment setups that used LDA and HMN.

Table 1 shows the impact of different word embedding models in vertex representation, with and without dimensionality reduction on POS, NER, and Chunking tasks in the CoNLL 2003 data set. It also shows the impact of n-gram concatenation, and dimensionality reduction . Results are reported for different labeled data set sizes, which demonstrate a low-resourced setting.

Since there were no pre-trained embeddings available for WANG2VEC, we trained from the first billion characters from Wikipedia for English. This lead to an sub optimal results across all tasks, hence we have omitted from reporting it.

As indicated by the results in Table 1, it is evident that ELMO performs much better than FASTTEXT for all the tasks and all the data set sizes. While n-gram concatenation or dimensionality reduction did not show compelling results when used in isolation, when combined they contributed to a significant performance gain for both FASTTEXT and ELMO.

In this experiment, we used Annoy approximate nearest neighbor algorithm to quickly calculate the nearest neighbors. Benchmarks done on ANN (Aumüller et al., 2017) have shown accuracy drops when the dimension increases above 100. This can be seen in our results - with concatenated vectors or high dimension vectors like ELMO the accuracy is considerably lower. Since our approach was transductive, we were wary of the efficiency and timing. Traditional k-NN algorithms gave better scores but lead to high time and memory consumption.

Tables 2 and 3 show the results of similar experiments carried out for Sinhala and Tamil POS tagging tasks, respectively. While FASTTEXT performs better than WANG2VEC for Tamil, the opposite was noted for Sinhala. We attribute this difference to the differences in the models created for the two languages - WANG2VEC and FAST-

	POS					Chunking					NER				
	20K	40K	60K	80K	100K	20K	40K	60K	80K	100K	20K	40K	60K	80K	100K
FastText															
A	0.75	0.79	0.83	0.839	0.81	0.66	0.70	0.73	0.73	0.71	0.35	0.30	0.46	0.46	0.34
B	0.69	0.72	0.74	0.77	0.74	0.66	0.69	0.67	0.72	0.74	0.31	0.25	0.44	0.43	0.35
C	0.60	0.64	0.68	0.70	0.66	0.53	0.57	0.57	0.69	0.60	0.38	0.30	0.44	0.46	0.39
D	0.85	0.88	0.87	0.88	0.86	0.79	0.83	0.85	0.83	0.83	0.61	0.53	0.69	0.66	0.50
ELMo															
A	0.84	0.84	0.88	0.88	0.86	0.82	0.85	0.85	0.82	0.84	0.70	0.67	0.84	0.81	0.65
B	0.90	0.91	0.92	0.92	0.91	0.82	0.83	0.84	0.83	0.84	0.69	0.65	0.76	0.79	0.70
C	0.74	0.76	0.83	0.81	0.77	0.76	0.80	0.79	0.78	0.78	0.62	0.56	0.81	0.77	0.57
D	0.928	0.934	0.941	0.942	0.93	0.90	0.91	0.92	0.88	0.90	0.79	0.76	0.86	0.89	0.70

Table 1: Comparison of different methods to represent nodes and their respective accuracy for different tasks in English. A - Single Vector, B - Dimension reduced Single Vector, C - Concatenated n-gram vectors, D - Dimension reduced concatenated n-gram vectors.

	Tamil POS			Sinhala POS				
	20K	40K	60K	20K	40K	60K	80K	100K
FastText								
A	0.77	0.81	0.73	0.80	0.76	0.83	0.82	0.77
B	0.62	0.79	0.77	0.80	0.77	0.83	0.82	0.79
C	0.54	0.58	0.58	0.66	0.60	0.67	0.66	0.59
D	0.87	0.88	0.89	0.901	0.88	0.88	0.85	0.84
Wang2Vec								
A	0.72	0.74	0.70	0.815	0.775	0.84	0.81	0.77
B	0.59	0.71	0.54	0.78	0.76	0.81	0.79	0.77
C	0.58	0.82	0.57	0.714	0.66	0.70	0.70	0.63
D	0.70	0.71	0.72	0.801	0.76	0.84	0.85	0.81

Table 2: Comparison of different methods to represent nodes and their respective accuracy for Tamil and Sinhala POS tagging. A - Single Vector, B - Dimension reduced Single Vector, C - Concatenated n-gram vectors, D - Dimension reduced concatenated n-gram vectors.

Text models for Sinhala were created using a much larger corpus than that for Tamil. Moreover, domain-similarity was much higher between the Sinhala test data and the data used to build the models. In line with the observation for English, for both the languages, FastText performs better when concatenated and dimensionality is reduced. However, contrary to our expectations, the same is not clearly observed with respect to Wang2Vec.

We then compared the performance of our GSSL approach against Peters et al. (2017) using the best result reported in Table 1. As shown in Figures 1, 2 and 3, when the ELMo model with n-gram concatenation and dimensionality reduction is used, our GSSL approach outperforms Peters et al. (2017)'s bidirectional LSTM CRF.

According to these Figures, when increasing training data, opposed to our expectations there are some drops in scores. One of the glaring one was with NER. For dimension reduced concatenated ELMo vector with 80K training data re-

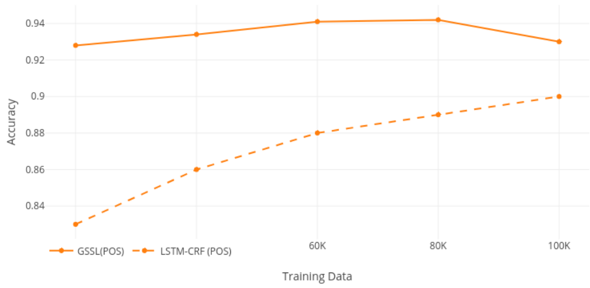

Figure 1: POS accuracy for GSSL Vs LSTM-CRF

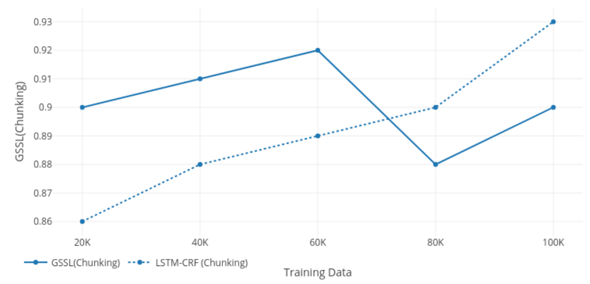

Figure 2: Chunking F1-Score for GSSL Vs LSTM-CRF

Figure 3: NER F1-Score for GSSL Vs LSTM-CRF

sulted an 0.89 F1 score, and it drops to 0.7 for 100K training data. Further analysis revealed that when training data set was increased, it had lead to over-fitting. For example our training data had *Germany* as *LOC* and the test data had *German* which was supposed to be classified as *MISC* was classified as *LOC* due to the close proximity of vectors.

Fernando et al. (2016) had presented POS tagger for Sinhala using hand crafted language dependent features. This research reported the best accuracy for the University of Moratuw corpus. We sampled out a 20K dataset form this corpus as training data for both ours and Fernando et al. (2016)'s approach. The SVM Tagger reported an accuray of 87.11% while we were able to achieve an accuracy of 90.1%. Mokanarangan et al. (2018) had reported for GSSL based approach for FIRE POS tagging with an accuracy of 87.43% for 60K data. For the same training data we were able to achieve an accuracy of 89%.

6 Conclusion

The aim of this research was to develop an efficient GSSL solution for sequence tagging. Our solution is based on identifying neural word embedding models that better capture local context information in graph vertices, and producing a graph in a low-dimensional space that has vertices belonging to different classes well-separated. While some of the word embedding models employed did not generate the expected result, in general, our hypothesis of capturing context information by concatenating vectors is validated. In particular, n-gram concatenation and dimensionality reduction resulted in significant performance gains. Given the fact that our best result outperforms the existing state-of-the-art (for high resource settings), when the labeled data set size is small, our GSSL solution can be presented as a promising alternative for sequence tagging in low-resourced languages.

In the current implementation, LDA calculations are done mostly in memory. Thus when we attempt to use larger annotated training sets with each vector having over 900 dimensions leads to memory overflows. Since our target was towards addressing low resource settings, we did not attempt to address this issue. Thus scalability of our approach for high resource settings should be explored with more optimal dimensionality reduction approaches.

References

Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull. 2017. Ann-benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. In *International Conference on Similarity Search and Applications*, pages 34–49. Springer.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 238–247.

E.: Annoy Bernhardsson. 2018. Annoy - Approximate Nearest Neighbor. https://github.com/spotify/annoy. [Online; accessed 21-Feb-2018].

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Semi-supervised learning for neural machine translation. *CoRR*, abs/1606.04596.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 600–609. Association for Computational Linguistics.

Saner Demirel. 2017. *Spectral Graph Convolutional Networks for Part-of-Speech Tagging*. Ph.D. thesis, Universität Koblenz-Landau.

Paramveer S Dhillon, Partha Pratim Talukdar, and Koby Crammer. 2010. Inference driven metric learning (idml) for graph construction.

Sandareka Fernando, Surangika Ranathunga, Sanath Jayasena, and Gihan Dias. 2016. Comprehensive part-of-speech tag set and svm based pos tagger for sinhala. In *Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing (WSSANLP2016)*, pages 173–182.

Dan Garrette, Jason Mielens, and Jason Baldridge. 2013. Real-world semi-supervised learning of pos-taggers for low-resource languages. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 583–592. Association for Computational Linguistics.

Luheng He, Jennifer Gillenwater, and Ben Taskar. 2013. Graph-based posterior regularization for semi-supervised structured prediction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 38–46.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Wang Ling, Chris Dyer, Alan W Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304.

Prasenjit Majumder, Mandar Mitra, Dipasree Pal, Ayan Bandyopadhyay, Samaresh Maiti, Sukanya Mitra, Aparajita Sen, and Sukomal Pal. 2008. Text collections for fire. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 699–700. ACM.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Rada F. Mihalcea and Dragomir R. Radev. 2011. *Graph-based Natural Language Processing and Information Retrieval*, 1st edition. Cambridge University Press, New York, NY, USA.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Thayaparan Mokanarangan, Ranathunga Surangika, and Thayasivam Uthayasanker. 2018. Graph based semi-supervised learning approach for tamil pos tagging. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. *CoRR*, abs/1705.00108.

François Rousseau and Michalis Vazirgiannis. 2013. Graph-of-word and tw-idf: New approach to ad hoc ir. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 59–68, New York, NY, USA. ACM.

François Rousseau, Emmanouil Kiagias, and Michalis Vazirgiannis. 2015. Text categorization as a graph classification problem. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1702–1712.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *CoRR*, cs.CL/0306050.

Konstantinos Skianis, François Rousseau, and Michalis Vazirgiannis. 2016. Regularizing text categorization with clusters of words. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1827–1837.

Amarnag Subramanya and Jeff Bilmes. 2008. Soft-supervised learning for text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 1090–1099, Stroudsburg, PA, USA. Association for Computational Linguistics.

Amarnag Subramanya, Slav Petrov, and Fernando Pereira. 2010. Efficient graph-based semi-supervised learning of structured tagging models. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 167–176, Stroudsburg, PA, USA. Association for Computational Linguistics.

Masashi Sugiyama. 2006. Local fisher discriminant analysis for supervised dimensionality reduction. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 905–912, New York, NY, USA. ACM.

Partha Pratim Talukdar and Fernando Pereira. 2010. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1473–1481, Stroudsburg, PA, USA. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. 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*, ConLL '00, pages 127–132, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. 2003. Learning with local and global consistency. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, pages 321–328, Cambridge, MA, USA. MIT Press.

Xiaojin Zhu. 2005. Semi-supervised learning literature survey.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pages 912–919. AAAI Press.

Bibliography

[1] Erik F Tjong Kim Sang and Fien De Meulder. 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, 2003.

[2] Beth M Sundheim. Overview of results of the muc-6 evaluation. In *Proceedings of the 6th conference on Message understanding*, pages 13–31. Association for Computational Linguistics, 1995.

[3] Diego Mollá, Menno Van Zaanen, Steve Cassidy, et al. Named entity recognition in question answering of speech data. 2007.

[4] Changki Lee, Yi-Gyu Hwang, Hyo-Jung Oh, Soojong Lim, Jeong Heo, Chung-Hee Lee, Hyeon-Jin Kim, Ji-Hyun Wang, and Myung-Gil Jang. Fine-grained named entity recognition using conditional random fields for question answering. In *Asia Information Retrieval Symposium*, pages 581–587. Springer, 2006.

[5] Einat Minkov, Richard C Wang, and William W Cohen. Extracting personal names from email: Applying named entity recognition to informal text. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 443–450. Association for Computational Linguistics, 2005.

[6] Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun, and Bu-Sung Lee. Twiner: named entity recognition in targeted twitter stream. In *Proceedings of the 35th international ACM SIGIR conference on*

Research and development in information retrieval, pages 721–730. ACM, 2012.

[7] Giridhar Kumaran and James Allan. Text classification and named entities for new event detection. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 297–304. ACM, 2004.

[8] Alireza Mansouri, Lilly Suriani Affendey, and Ali Mamat. Named entity recognition approaches. *International Journal of Computer Science and Network Security*, 8(2):339–344, 2008.

[9] James R Curran and Stephen Clark. Language independent ner using a maximum entropy tagger. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 164–167. Association for Computational Linguistics, 2003.

[10] Rahul Sharnagat. Named entity recognition: A literature survey. *Center For Indian Language Technology*, 2014.

[11] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[12] Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. *CoRR*, abs/1705.00108, 2017. URL `http://arxiv.org/abs/1705.00108`.

[13] Pranavan Theivendiram, Megala Uthayakumar, Nilusija Nadarasamoorthy, Mokanarangan Thayaparan, Sanath Jayasena, Gihan Dias, and Surangika Ranathunga. Named-entity-recognition (ner) for tamil language using margin-infused relaxed algorithm (mira). In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 465–476. Springer, 2016.

[14] SAPM Manamini, AF Ahamed, RAEC Rajapakshe, GHA Reemal, S Jayasena, GV Dias, and S Ranathunga. Ananya-a named-entity-recognition (ner) system for sinhala language. In *Moratuwa Engineering Research Conference (MERCon), 2016*, pages 30–35. IEEE, 2016.

[15] JK Dahanayaka and AR Weerasinghe. Named entity recognition for sinhala language. In *Advances in ICT for Emerging Regions (ICTer), 2014 International Conference on*, pages 215–220. IEEE, 2014.

[16] R Vijayakrishna and L Sobha. Domain focused named entity recognizer for tamil using conditional random fields. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*, 2008.

[17] Dan Garrette, Jason Mielens, and Jason Baldridge. Real-world semi-supervised learning of pos-taggers for low-resource languages. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 583–592. Association for Computational Linguistics, 2013. URL `http://www.aclweb.org/anthology/P13-1057`.

[18] Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Semi-supervised learning for neural machine translation. *CoRR*, abs/1606.04596, 2016. URL `http://arxiv.org/abs/1606.04596`.

[19] Amarnag Subramanya and Jeff Bilmes. Soft-supervised learning for text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 1090–1099, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1613715.1613857`.

[20] Mouiad Fadiel Alawneh and Tengku Mohd Sembok. Rule-based and example-based machine translation from english to arabic. In *Bio-Inspired Computing: Theories and Applications (BIC-TA), 2011 Sixth International Conference on*, pages 343–347. IEEE, 2011.

[21] Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. A statistical approach to machine translation. *Computational linguistics*, 16 (2):79–85, 1990.

[22] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. URL `http://arxiv.org/abs/1409.0473`.

[23] Pasindu Tennage, Prabath Sandaruwan, Malith Thilakarathne, Achini Herath, and Surangika Ranathunga. Handling Rare Word Problem using Synthetic Training Data for Sinhala and Tamil Neural Machine Translation. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018 2018. European Language Resources Association (ELRA). ISBN 979-10-95546-00-9.

[24] Fathima Farhath, Pranavan Theivendiram, Surangika Ranathunga, Sanath Jayasena, and Gihan Dias. Improving domain-specific SMT for low-resourced languages using data from different domains. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018 2018. European Language Resources Association (ELRA). ISBN 979-10-95546-00-9.

[25] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics, 2007.

[26] Behrang Mohit. Named entity recognition. In *Natural language processing of semitic languages*, pages 221–245. Springer, 2014.

[27] Hristo Tanev, Jakub Piskorski, and Martin Atkinson. Real-time news event extraction for global crisis monitoring. In *International Conference on Application of Natural Language to Information Systems*, pages 207–218. Springer, 2008.

[28] Sriparna Saha, Asif Ekbal, and Utpal Kumar Sikdar. Named entity recognition and classification in biomedical text using classifier ensemble. *International journal of data mining and bioinformatics*, 11(4):365–391, 2015.

[29] Prasenjit Majumder, Mandar Mitra, Dipasree Pal, Ayan Bandyopadhyay, Samaresh Maiti, Sukanya Mitra, Aparajita Sen, and Sukomal Pal. Text collections for fire. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 699–700. ACM, 2008.

[30] Douglas E Appelt, Jerry R Hobbs, John Bear, David Israel, Megumi Kameyama, David Martin, Karen Myers, and Mabry Tyson. Sri international fastus system: Muc-6 test results and analysis. In *Proceedings of the 6th conference on Message understanding*, pages 237–248. Association for Computational Linguistics, 1995.

[31] Ralph Grishman. Information extraction: Techniques and challenges. In *International Summer School on Information Extraction*, pages 10–27. Springer, 1997.

[32] Michal Konkol and Miloslav Konopík. Named entity recognition for highly inflectional languages: effects of various lemmatization and stemming approaches. In *International Conference on Text, Speech, and Dialogue*, pages 267–274. Springer, 2014.

[33] Daniel M Bikel, Richard Schwartz, and Ralph M Weischedel. An algorithm that learns what's in a name. *Machine learning*, 34(1-3):211–231, 1999.

[34] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3): 268–278, 1973.

[35] GuoDong Zhou and Jian Su. Named entity recognition using an hmm-based chunk tagger. In *proceedings of the 40th Annual Meeting on Association for*

Computational Linguistics, pages 473–480. Association for Computational Linguistics, 2002.

[36] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[37] Label bias problem. `https://cs.nyu.edu/courses/spring13/CSCI-GA.2590-001/LabelBias.pptx`. Accessed: 2018-06-17.

[38] Paul McNamee and James Mayfield. Entity extraction without language-specific resources. In *proceedings of the 6th conference on Natural language learning-Volume 20*, pages 1–4. Association for Computational Linguistics, 2002.

[39] Sriparna Saha and Asif Ekbal. Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition. *Data & Knowledge Engineering*, 85:15–39, 2013.

[40] Amarnag Subramanya, Slav Petrov, and Fernando Pereira. Efficient graph-based semi-supervised learning of structured tagging models. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 167–176, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1870658.1870675`.

[41] Partha Pratim Talukdar and Fernando Pereira. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1473–1481, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1858681.1858830`.

[42] Xavier Carreras, Lluís Màrquez, and Lluís Padró. A simple named entity extractor using adaboost. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 152–155. Association for Computational Linguistics, 2003.

[43] Gunnar Rätsch, Takashi Onoda, and K-R Müller. Soft margins for adaboost. *Machine learning*, 42(3):287–320, 2001.

[44] Amarnag Subramanya and Partha Pratim Talukdar. Graph-based semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(4):1–125, 2014.

[45] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[46] Enrique Alfonseca and Suresh Manandhar. An unsupervised method for general named entity recognition and automated concept discovery. In *Proceedings of the 1st international conference on general WordNet, Mysore, India*, pages 34–43, 2002.

[47] Yusuke Shinyama and Satoshi Sekine. Named entity discovery using comparable news articles. In *Proceedings of the 20th international conference on Computational Linguistics*, page 848. Association for Computational Linguistics, 2004.

[48] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134, 2005.

[49] Ryan Cotterell and Kevin Duh. Low-resource named entity recognition with cross-lingual, character-level neural conditional random fields. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 91–96, 2017.

[50] Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1946–1958, 2017.

[51] Zhilin Yang, Ruslan Salakhutdinov, and William Cohen. Multi-task cross-lingual sequence tagging from scratch. *arXiv preprint arXiv:1603.06270*, 2016.

[52] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[53] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

[54] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*, 2016.

[55] Nils Reimers and Iryna Gurevych. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *arXiv preprint arXiv:1707.06799*, 2017.

[56] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[57] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[58] Tomáš Mikolov, Anoop Deoras, Daniel Povey, Lukáš Burget, and Jan Černockỳ. Strategies for training large scale neural network language models. In *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, pages 196–201. IEEE, 2011.

[59] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[60] CS Malarkodi, RK Pattabhi, and Lalitha Devi Sobha. Tamil ner–coping with real time challenges. In *24th International Conference on Computational Linguistics*, page 23, 2012.

[61] Wei Li and Andrew McCallum. Rapid development of hindi named entity recognition using conditional random fields and feature induction. *ACM*

Transactions on Asian Language Information Processing (TALIP), 2(3): 290–294, 2003.

[62] Sujan Kumar Saha, Sudeshna Sarkar, and Pabitra Mitra. A hybrid feature set based maximum entropy hindi named entity recognition. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*, 2008.

[63] Anup Patel, Ganesh Ramakrishnan, and Pushpak Bhattacharya. Relational learning assisted construction of rule base for indian language ner. *Proceedings of ICON*, 2009:7th, 2009.

[64] Karthik Gali, Harshit Surana, Ashwini Vaidya, Praneeth Shishtla, and Dipti Misra Sharma. Aggregating machine learning and rule based heuristics for named entity recognition. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*, 2008.

[65] Animesh Nayan, B Ravi Kiran Rao, Pawandeep Singh, Sudip Sanyal, and Ratna Sanyal. Named entity recognition for indian languages. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*, 2008.

[66] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[67] Hamish Cunningham, Valentin Tablan, Angus Roberts, and Kalina Bontcheva. Getting more out of biomedical documents with gate's full lifecycle open source text analytics. *PLoS computational biology*, 9(2): e1002854, 2013.

[68] Steven Bird and Edward Loper. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics, 2004.

[69] Ralph Grishman and Beth Sundheim. Design of the muc-6 evaluation. In *Proceedings of the 6th conference on Message understanding*, pages 1–11. Association for Computational Linguistics, 1995.

[70] Rada F. Mihalcea and Dragomir R. Radev. *Graph-based Natural Language Processing and Information Retrieval*. Cambridge University Press, New York, NY, USA, 1st edition, 2011. ISBN 0521896134, 9780521896139.

[71] Xiaojin Zhu, John Lafferty, and Ronald Rosenfeld. *Semi-supervised learning with graphs*. PhD thesis, Carnegie Mellon University, language technologies institute, school of computer science, 2005.

[72] Hany Hassan and Arul Menezes. Social text normalization using contextual graph random walks. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1577–1586, 2013.

[73] Linhong Zhu, Sheng Gao, Sinno Jialin Pan, Haizhou Li, Dingxiong Deng, and Cyrus Shahabi. Graph-based informative-sentence selection for opinion summarization. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '13, pages 408–412, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2240-9. doi: 10.1145/2492517.2492651. URL `http://doi.acm.org/10.1145/2492517.2492651`.

[74] Micha Elsner and Eugene Charniak. Disentangling chat. *Comput. Linguist.*, 36(3):389–409, September 2010. ISSN 0891-2017. doi: 10.1162/coli_a_00003. URL `http://dx.doi.org/10.1162/coli_a_00003`.

[75] Dipanjan Das and Slav Petrov. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 600–609. Association for Computational Linguistics, 2011.

[76] Luheng He, Jennifer Gillenwater, and Ben Taskar. Graph-based posterior regularization for semi-supervised structured prediction. In *Proceedings of*

the Seventeenth Conference on Computational Natural Language Learning, pages 38–46, 2013.

[77] Saner Demirel. *Spectral Graph Convolutional Networks for Part-of-Speech Tagging*. PhD thesis, Universität Koblenz-Landau, 2017.

[78] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 238–247, 2014.

[79] Stefan Evert. The statistics of word cooccurrences: word pairs and collocations. 2005.

[80] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. 16 January 2013.

[81] Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.

[82] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

[83] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[84] Wang Ling, Chris Dyer, Alan W Black, and Isabel Trancoso. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304, 2015.

[85] Language models, word2vec, and efficient softmax approximations. `http://rohanvarma.me/Word2Vec/`. Accessed: 2018-06-17.

[86] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. ISSN 2307-387X.

[87] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*, 2017.

[88] Chris Callison-Burch, David Talbot, and Miles Osborne. Statistical machine translation with word-and sentence-aligned parallel corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 175. Association for Computational Linguistics, 2004.

[89] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics, 2003.

[90] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. URL `http://arxiv.org/abs/1609.08144`.

[91] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[92] Krzysztof Wołk and Danijel Koržinek. Comparison and adaptation of automatic evaluation metrics for quality assessment of re-speaking. *arXiv preprint arXiv:1601.02789*, 2016.

[93] George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145. Morgan Kaufmann Publishers Inc., 2002.

[94] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51, 2003.

[95] Ruvan Weerasinghe. A statistical machine translation approach to sinhala-tamil language translation. *Towards an ICT enabled Society*, page 136, 2003.

[96] Roni Rosenfeld and Philip Clarkson. Statistical language modeling using the cmu-cambridge toolkit. 1997.

[97] Sakthithasan Sripirakas, AR Weerasinghe, and Dulip L Herath. Statistical machine translation of systems for sinhala-tamil. In *Advances in ICT for Emerging Regions (ICTer), 2010 International Conference on*, pages 62–68. IEEE, 2010.

[98] Andreas Stolcke. Srilm-an extensible language modeling toolkit. In *Seventh international conference on spoken language processing*, 2002.

[99] Language technology research lab - university of colombo school of computing. `http://ucsc.cmb.ac.lk/ltrl/projects/`. Accessed: 2018-06-17.

[100] Randil Pushpananda, Ruvan Weerasinghe, and Mahesan Niranjan. Sinhala-tamil machine translation: Towards better translation quality. In *Proceedings of the Australasian Language Technology Association Workshop 2014*, pages 129–133, 2014.

[101] S Rajpirathap, S Sheeyam, K Umasuthan, and Amalraj Chelvarajah. Real-time direct translation system for sinhala and tamil languages. In *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*, pages 1437–1443. IEEE, 2015.

[102] Randil Pushpananda, Ruvan Weerasinghe, and Mahesan Niranjan. Statistical machine translation from and into morphologically rich and low

resourced languages. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 545–556. Springer, 2015.

[103] Mathias Creutz and Krista Lagus. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(1):3, 2007.

[104] Fei Huang and Stephan Vogel. Improved named entity translation and bilingual named entity extraction. In *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*, pages 253–258. IEEE, 2002.

[105] Reinhard Rapp. Automatic identification of word translations from unrelated english and german corpora. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 519–526. Association for Computational Linguistics, 1999.

[106] Stephen Wan and Cornelia Maria Verspoor. Automatic english-chinese name transliteration for development of multilingual resources. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 1352–1356. Association for Computational Linguistics, 1998.

[107] GAO Wei. Phoneme based statistical transliteration of foreign names for oov problem. *Master's Thesis, The Chinese University of Hong Kong*, 2004.

[108] Paola Virga and Sanjeev Khudanpur. Transliteration of proper names in cross-lingual information retrieval. In *Proceedings of the ACL 2003 workshop on Multilingual and mixed-language named entity recognition-Volume 15*, pages 57–64. Association for Computational Linguistics, 2003.

[109] Long Jiang, Ming Zhou, Lee-Feng Chien, and Cheng Niu. Named entity translation with web mining and transliteration. In *IJCAI*, volume 7, pages 1629–1634, 2007.

[110] Asif Ekbal, Sudip Kumar Naskar, and Sivaji Bandyopadhyay. Named entity transliteration. *International Journal of Computer Processing of Oriental Languages*, 20(04):289–310, 2007.

[111] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[112] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, 2013.

[113] Paramveer S Dhillon, Partha Pratim Talukdar, and Koby Crammer. Inference driven metric learning (idml) for graph construction. 2010.

[114] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pages 912–919. AAAI Press, 2003. ISBN 1-57735-189-4. URL `http://dl.acm.org/citation.cfm?id=3041838.3041953`.

[115] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

[116] Open tamil. `https://github.com/Ezhil-Language-Foundation/open-tamil`. Accessed: 2018-06-17.

[117] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[118] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[119] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[120] Timothy Dozat. Incorporating nesterov momentum into adam. 2016.

[121] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[122] Tomáš Mikolov. Statistical language models based on neural networks. *Presentation at Google, Mountain View, 2nd April*, 2012.

[123] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

[124] Radu Soricut and Franz Och. Unsupervised morphology induction using word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1627–1637, 2015.

[125] Juan Miguel Cejuela, Peter McQuilton, Laura Ponting, Steven J Marygold, Raymund Stefancsik, Gillian H Millburn, and Burkhard Rost. tagtog: interactive and text-mining-assisted annotation of gene mentions in plos full-text articles. *Database*, 2014, 2014.

[126] Allennlp. `https://github.com/allenai/allennlp`. Accessed: 2018-06-17.

[127] Uom allen nlp repo. `https://github.com/Mokanarangan/UOM-Alle`. Accessed: 2018-06-28.

[128] Tensorflow implementation of contextualized word representations from bidirectional language models. `https://github.com/allenai/bilm-tf`. Accessed: 2018-06-17.

[129] Language models. `https://web.stanford.edu/class/cs124/lec/languagemodeling.pdf`. Accessed: 2018-06-17.

[130] Pre-trained word vectors. `https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md`. Accessed: 2018-06-17.

[131] Wang2vec. `https://github.com/wlin12/wang2vec`. Accessed: 2018-06-17.

[132] metric-learn: Metric learning in python. `http://metric-learn.github.io/metric-learn/`. Accessed: 2018-06-17.

[133] E.: Annoy Bernhardsson. Annoy - Approximate Nearest Neighbor. `https://github.com/spotify/annoy`, 2018. [Online; accessed 21-Feb-2018].

[134] Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull. Ann-benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. In *International Conference on Similarity Search and Applications*, pages 34–49. Springer, 2017.

[135] Masashi Sugiyama. Local fisher discriminant analysis for supervised dimensionality reduction. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 905–912, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143958. URL `http://doi.acm.org/10.1145/1143844.1143958`.

[136] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, pages 321–328, Cambridge, MA, USA, 2003. MIT Press. URL `http://dl.acm.org/citation.cfm?id=2981345.2981386`.

[137] Moses. `http://www.statmt.org/moses/?n=Advanced.Hybrid`. Accessed: 2018-06-17.

[138] Sandareka Fernando, Surangika Ranathunga, Sanath Jayasena, and Gihan Dias. Comprehensive part-of-speech tag set and svm based pos tagger for sinhala. In *Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing (WSSANLP2016)*, pages 173–182, 2016.

[139] Kenneth Heafield. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics, 2011.

[140] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics, 2003.

[141] Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. *arXiv preprint arXiv:1603.08861*, 2016.