Deliverable 3.1 (placeholder for your finished Viterbi table in Section 3)

	they	can	can	fish	END
Start	n/a	n/a	n/a	n/a	n/a
Noun	-2	-10	-10	-15	n/a
Verb	-13	-6	-11	-16	n/a
End	n/a	n/a	n/a	n/a	-17

Deliverable 8.1 (4 points):

It is rather straightforward to see why our initial Naive Bayes tagger is not a good sequence labeling model since it considers each word in the sequence separately. However, unlike Naive Bayes, LSTMs encode the left context of each word. I.e., the model is aware of the hidden states of previous words when deciding the POS tag for the current word.

Is the CRF still useful to LSTMs? Do you think CRF would make a greater performance improvement for high-resource languages (e.g., English) or low-resource languages (e.g., Tamil)? Why? What types of tasks might benefit more or less from adding the CRF?

CRF is still useful to LSTM as it is currently used with LSTMs for sequence labelling tasks. CRF capture information about the neighbours in a sequence and in turn captures more information and is therefore more useful for training. I think CRF will make a greater performance improvement for high-resource languages as compared low resource languages. For sequence labeling (or general structured prediction) tasks, it is intuitive to consider that there may be a relation between neighbouring words, for example we know that adjective is more likely to be followed by noun, therefore clearly decoding each label separately for each token in a sequence should not be preferred. In high resource language, CRF will perform better as it will capture more information about the neighbouring tokens and will consequently perform better.

Therefore, clearly, CRFs should be preferred in sequential tasks(where there is a relation between the neighbours) such as sequence labelling tasks. On the other hand, in cases, where there is no relation between the neighbouring entities, CRF may not be very useful.

Deliverable 8.2 (4 points):

Briefly describe your bakeoff design.

For English Language

For English language, I used Polyglot word embeddings. I used a BiLSTM with embedding dimension = 64, hidden dimension as 80 and number of layers as 3. I used learning rate of 0.05 and performed 60 iterations. I achieved an accuracy fo 87.13% on the dev set. I performed parameter tuning to achieve these results.

Norwegian Language

For Norwegian, I used Polyglot word embeddings. I used a BiLSTM with embedding dimension = 64 and hidden dimension as 40. I used learning rate of 0.4 and performed 20 iterations. I achieved an accuracy fo 88.27% on the dev set. I performed parameter tuning to achieve these results.

Deliverable 8.3 (8 points):

You will select a research paper at ACL, EMNLP or NAACL that performs **sequence labeling**. Summarize the paper, answering the following questions:

- **List the title, author(s) and venue of the paper.**

  Paper title - End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF.

  Authors - Xuezhe Ma and Eduard Hovy, Carnegie Mellon University

  Venue - Annual Meeting of the Association for Computational Linguistics (2016)

- **What is the task they are trying to solve?**

Problem - The current state-of-the-art sequence labeling systems such as Hidden Markov Models (HMM) and Conditional Random Fields (CRF) require large amounts of task-specific knowledge in the form of handcrafted features and data pre-processing. This task-specific knowledge can be costly to develop making sequence labeling models difficult to adapt to new tasks and new domains.

Proposed solution - The authors introduce a novel neural network architecture that learns the task-specific features automatically. They design an ended-to-end system that requires no hand-crafted features. They use a combination of bidirectional LSTM, CNN and CRF to allow the network to learn from both word- and character-level representations automatically. Authors evaluated their system on two data sets for two sequence labeling tasks — Penn Treebank WSJ corpus for part-of-speech (POS) tagging

and CoNLL 2003 corpus for named entity recognition (NER). Authors showed that their system achieved state-of-the-art performance on both datasets.

- **What tagging methods do they use? E.g., HMMs, CRF, max-margin markov networks, deep learning models?**

Authors used a combination of bidirectional LSTM, CNN and CRF for the sequence tagging task. They designed an end-to-end system called 'Bi-directional LSTM-CNNs-CRF'.

Steps followed by authors:

- They first used a convolutional neural networks (CNNs) to encode character-level information of a word into its character-level representation. The CNN takes character embeddings as inputs and in turn produces the character-level representation for each word.
- This character-level representation vector is concatenated with the pre-trained word embedding vector. The word embedding vector was obtained using Stanfrd's GloVe embeddings. This combined vector is fed as input to BLSTM(bidirectional LSTM) network.
- The output vectors of BLSTM are fed to the CRF layer to jointly decode the labels for the entire sequence.

- **What features do they use and why?**

  Authors provided an end to end system 'Bi-directional LSTM-CNNs-CRF' that required no task-specific resources, feature engineering, or data pre-processing beyond pre-trained word embeddings on unlabeled corpora. Authors mention that developing handcrafted features for tagging tasks can be expensive and time consuming, therefore, authors designed CNN network which learned the features automatically. The features learnt by CNN were concatenated with pretrained embeddings from Stanford's publicly available GloVe (100-dimensional embeddings which are trained on 6 billion words from Wikipedia and web text).

  The authors performed an experiment to decide on which pretrained embeddings should be used for concatenation with CNN features. In addition to Standford 's GloVe embeddings, authors also considered: - Senna 50- dimensional embeddings trained on Wikipedia and Reuters RCV-1 corpus (Collobert et al., 2011), Google's Word2Vec 300-dimensional embeddings trained on 100 billion words from Google News (Mikolov et al., 2013) and randomly initialized embeddings with 100 dimensions. Authors found that Stanford' GloVe performed

the best with their proposed Bi-directional LSTM-CNNs-CRF architecture and decided to go ahead with it.

- **What methods and features are most effective?**

  Authors show that their proposed architecture 'Bi-directional LSTM-CNNs-CRF' gives comparable performance to the state-of-art sequence tagging methods. The tables below show a comparison of the accuracy and F1 scores achieved by the proposed algorithm and the state-of-art algorithms on the Penn Treebank WSJ. Additionally, this method required no task-specific resources, feature engineering, or data pre-processing beyond pre-trained word embeddings on unlabeled corpora.

Model	Acc.
Giménez and Màrquez (2004)	97.16
Toutanova et al. (2003)	97.27
Manning (2011)	97.28
Collobert et al. (2011)‡	97.29
Santos and Zadrozny (2014)‡	97.32
Shen et al. (2007)	97.33
Sun (2014)	97.36
Søgaard (2011)	97.50
This paper	**97.55**

Table 4: POS tagging accuracy of our model on test data from WSJ proportion of PTB, together with top-performance systems. The neural network based models are marked with ‡.

Model	F1
Chieu and Ng (2002)	88.31
Florian et al. (2003)	88.76
Ando and Zhang (2005)	89.31
Collobert et al. (2011)‡	89.59
Huang et al. (2015)‡	90.10
Chiu and Nichols (2015)‡	90.77
Ratinov and Roth (2009)	90.80
Lin and Wu (2009)	90.90
Passos et al. (2014)	90.90
Lample et al. (2016)‡	90.94
Luo et al. (2015)	91.20
This paper	**91.21**

These tables indicate that the proposed algorithm gives comparable performance to state-of-art methods for POS tagging.

Most effective features - Experiments by the author show that allowing the network to learn the feature automatically is an effective way for POS tagging

and named entity recognition. Author's experiments also show the effectiveness of using Stanford's GloVe embeddings for concatenation with CNN output over other embeddings for sequence labelling tasks.

- **Give a one-line summary of the paper that the authors are trying to leave for the reader.**

Most traditional high performance sequence labeling models rely heavily on hand-crafted features and taskspecific resources which are costly to develop, therefore authors propose a novel Bi-directional LSTM-CNNs-CRF architecture which performs end-to-end sequence labelling.