Homework 3: Neural word sense disambiguation

Task description and base model overview

Natural language comprises a huge amount of words which have multiple meanings and are therefore ambiguous. The human brain is very capable at solving such ambiguities, while for a machine it may be a very though challenge.
In the NLP community, word sense disambiguation (WSD) is the task of automatically selecting the most appropriate sense for a given word in a given context, be it a sentence or a whole document, among all the possible senses which can be associated to that word for the particular part-of-speech (POS) tag considered[1].
Traditionally, WSD has been tackled via many approaches, some of them being: always using the most frequent sense (MFS); sense with the greatest overlap of definitions w.r.t. a context window (Lesk [1]); sense with the highest score given by a random graph walk (or personalized PageRank) when exploiting a knowledge graph (UKB [2]).
Recently, with the advent of machine learning, and more specifically with deep learning, neural network approaches have become increasingly popular: typically, neural WSD is performed as single classification tasks for each word, considering a sliding window over the context; although quite efficient, this approach doesn't explicitly consider the sequential nature of sense distributions over the context.
An example of directly addressing such sequential nature of senses can be the one presented in *Neural Sequence Learning Models for Word Sense Disambiguation* [3], which is the reference paper of this homework assignment. In the reference paper, the authors suggest to tackle WSD as means of sequence tagging, possibly exploiting multi-task learning by also making the model perform related tasks like POS tagging.
This homework assignment has the target of performing WSD at different levels: other than over the fine-grained sense inventory of WordNet (which has been mapped to BabelSynsets), it is also required to implement coarse-grained WSD over WordNet Domains[2] and Lexicographer's names (Lexnames)[3], both of them being mapped to BabelSynsets as well.

Data preprocessing

The dataset used for this homework is SemCor, from the *Unified WSD Evaluation Framework* by Raganato et al. [4].
A first step in the preprocessing phase has been building input and output vocabularies from SemCor: the input vocabulary has been built by excluding all words with less than 5 occurrences and applying a subsapling equal to 10^{-4}; both these countermeasures are not applied for instance words, which have always been kept in the vocabulary.
The output vocabulary has been built by parsing the gold.key.txt file, and mapping the WordNet sense key to its associated BabelNet ID, WordNet Domain, or Lexname by using the provided mappings to BabelNet.
Another important step has been converting gold.key.txt files for every evaluation dataset (Senseval2, Senseval3, SemEval2007, SemEval2013, SemEval2015) to their BabelNet ID, WordNet Domains, and Lexnames versions in order to make use of the scorer provided by the unified framework.

Implementation

The proposed models are three, in multiple configurations each: BasicTagger, MultitaskTagger, and DomainAwareTagger. All of them are built upon a bidirectional LSTM with multiple layers, implementing dropout.
The BasicTagger model is the less complex of the three, which performs WSD over fine-grained senses only. Following the reference paper, the MultitaskTagger can be configured to also exploit POS tags and coarse-grained senses for the WSD task, still giving fine-grained senses as output. In this report, the MultitaskTagger is presented in several configurations: augmented with POS tags, WordNet Domains, or Lexnames (singularly) and combinations of POS tags and the two coarse-grained sense inventories.
The DomainAwareTagger is built upon the best performing MultitaskTagger configuration (see Evaluation paragraph), and it also takes into account the coarse-grained sense associated to the whole sentence: the intuition behind is that by knowing a whole sentence's domain should help by determining senses for words in that sentence.

[1] Ambiguity only arises when a word has multiple meanings for the same POS tag, since having multiple meanings each with a distinct POS tag is a very strong hint in order to distinguish the most appropriate meaning for the given context
[2] http://wndomains.fbk.eu/
[3] https://wordnet.princeton.edu/documentation/lexnames5wn

It is implemented by adding an extra embedding layer for the sentence's domain, and considering the last hidden state of the bidirectional LSTM; the cosine distance between the former and latter vectors is added to the loss function, as depicted in Figure 1.
In the DomainAwareTagger, bidirectional LSTM layers can be configured in order to exploit hidden state from previous layers in the swapped order[4]: the model initializes upper layers of the forward cell with the hidden state of the lower layer's backward cell (and viceversa). By configuring the model in this way, upper layers might exploit the internal representation of the whole sentence in the reversed order coming from the lower layers, possibly better addressing long-term dependencies between senses in the sentence.
For simplicity, the sentence domain inventory is the same inventory of the coarse-grained senses, and the sentence domain is chosen as the most frequent coarse-grained domain among the ones associated to the input sentence.
All the proposed models perform fine-grained WSD over BabelNet IDs, and the coarse-grained WSD is implemented as an additional mapping from BabelNet IDs to either WordNet Domains or Lexnames, as needed.

Evaluation

Given the abundance of evaluation datasets and following other works in the literature, SemEval2007 has been chosen as development set.
Hyper-parameter tuning has been performed via a grid search approach[5] (see Table 1) and the best tuning has been found by selecting the highest accuracy score recorded on the development set, which has been recorded to be 91.12%[6].
Once set, hyper-parameters have stayed the same for every other model and they have all been trained for 100 epochs on the full SemCor. Results are shown in Tables 2, 3, 4: values in **bold** are the best recorded scores, while values in *italic* are the runner up scores.
The best performing models can be identified in the following two: MultitaskTagger with Lexnames, and MultitaskTagger with POS tags and WordNet Domains. They are tied when it comes to fine-grained WSD over BabelNet IDs, but when considering coarse-grained WSD over WordNet Domains or over Lexnames, the best performing models are MultitaskTagger with Lexnames, and MultitaskTagger with POS tags and WordNet Domains (respectively).
Since WordNet Domains contains the semantically empty *factotum* class, higher scores for this coarse-grained WSD task can be slightly overlooked in favour of the one over Lexnames, and therefore the best performing model can be recognized in the MultitaskTagger with POS tags and WordNet Domains as coarse-grained sense inventory. This is also the model used as basis for the DomainAwareTagger.

Final considerations

The very simple approach implemented in MultitaskTagger with POS tags seems quite effective and raises some questions on how much value coarse-grained sense inventories really add to this model.
The DomainAwareTagger model could be extended in quite a few ways: a better sentence domain selection algorithm, and decoupling the coarse-grained sense inventory for the sentence domain from the one used for tagging each word in the sentence. Furthermore, given the architecture, the embedding layer for sentence domains could be replaced with pre-trained domain embeddings, potentially boosting its performances and overtaking other models presented here.

[4] $\text{DomainAware}_{\text{swap}}$ in Tables 2, 3, 4

[5] A BasicTagger model has been trained for 20 epochs over the first 50 batches of SemCor and tested against the first 15 batches of SemEval2007; batch size is equal to 16

[6] This accuracy score must not be mistaken for the precision or F1 score given by the Raganato et al. scorer

Tables and graphs

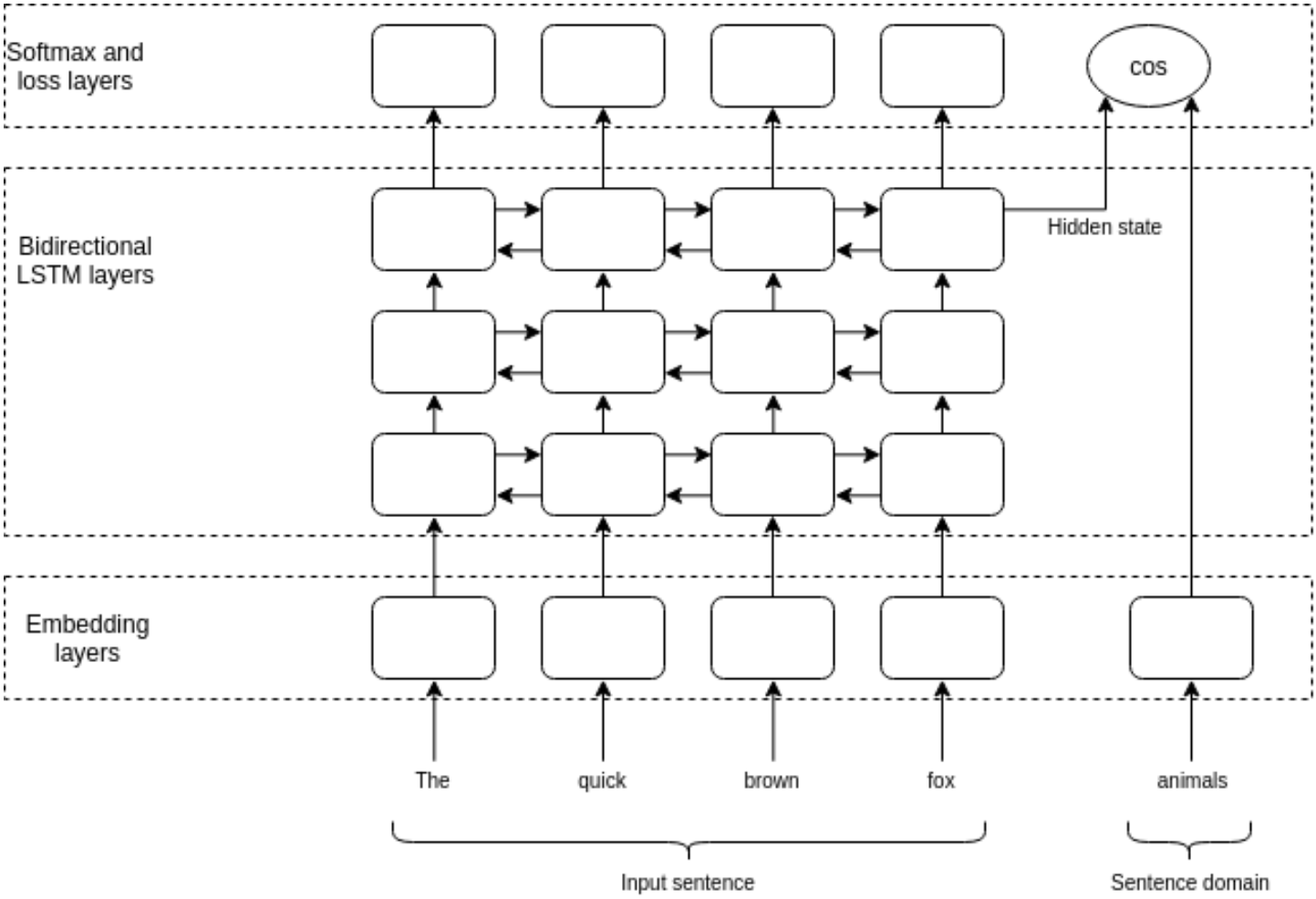

Figure 1: DomainAwareTagger model architecture

Hyper-parameter	Set of values	Best tuning
Embedding size	32, 64	64
Hidden size	32, 64	32
Learning rate	0.10, 0.15	0.10
Number of layers	1, 2	2

Table 1: Grid search used for hyper-parameter tuning and best scoring settings.

Model	SemEval2007	Senseval2	Senseval3	SemEval2013	SemEval2015	Average
MFS (baseline)	37.1	50.6	48.3	63.1	49.7	49.8
Basic	44.2	56.6	54	53.4	59	53.4
$\text{Multitask}_{\text{POS}}$	46.6	59.7	55	57.8	64	*56.6*
$\text{Multitask}_{\text{wnDom}}$	43.3	60.7	55.5	57	63.1	55.9
$\text{Multitask}_{\text{LEX}}$	41.8	63	58	59.2	63.6	**57.1**
$\text{Multitask}_{\text{POS + wnDom}}$	46.2	62.6	57.8	57.9	60.8	**57.1**
$\text{Multitask}_{\text{POS + LEX}}$	42.9	62.8	57.2	55.2	63.2	56.2
$\text{DomainAware}_{\text{no swap}}$	41.8	54.5	52.9	51.6	58.5	51.9
$\text{DomainAware}_{\text{swap}}$	44.2	56.3	56.1	58.2	58.4	54.6

Table 2: F1 scores for fine-grained WSD over BabelNet IDs.

Model	SemEval2007	Senseval2	Senseval3	SemEval2013	SemEval2015	Average
MFS (baseline)	71.9	77.7	71.4	76.6	71	73.7
Basic	81.3	85.9	80.8	70.3	80.6	79.8
$\text{Multitask}_{\text{POS}}$	83.7	88.4	82.5	74.4	84.6	*82.7*
$\text{Multitask}_{\text{wnDom}}$	82.2	89	81.9	73.7	83.5	82.1
$\text{Multitask}_{\text{LEX}}$	83.5	89.7	83.2	74.3	83.8	**82.9**
$\text{Multitask}_{\text{POS + wnDom}}$	84.6	90	83.7	72.5	82.7	*82.7*
$\text{Multitask}_{\text{POS + LEX}}$	81.5	90.3	83.6	72.6	83.3	82.2
$\text{DomainAware}_{\text{no swap}}$	81.8	85.6	79.7	69.5	81.5	79.6
$\text{DomainAware}_{\text{swap}}$	80.9	83.9	80.6	72.3	77.8	79.1

Table 3: F1 scores for coarse-grained WSD over WordNet Domains.

Model	SemEval2007	Senseval2	Senseval3	SemEval2013	SemEval2015	Average
MFS (baseline)	43.7	59.5	55.6	75.2	57.3	58.3
Basic	65.5	75.4	71.5	67.9	72.7	70.6
$\text{Multitask}_{\text{POS}}$	64.8	78.9	72.9	71.9	77.5	73.2
$\text{Multitask}_{\text{wnDom}}$	63.7	79.6	73	70.5	78.7	73.1
$\text{Multitask}_{\text{LEX}}$	62.6	80.7	72.9	72.3	78	*73.3*
$\text{Multitask}_{\text{POS + wnDom}}$	64.8	81.1	74.2	71.4	76.2	**73.5**
$\text{Multitask}_{\text{POS + LEX}}$	65.7	80.8	73.4	68.9	76.7	73.1
$\text{DomainAware}_{\text{no swap}}$	62.2	76	71	65.6	73.4	69.6
$\text{DomainAware}_{\text{swap}}$	60.9	71.9	70.1	70.9	70.5	68.9

Table 4: F1 scores for coarse-grained WSD over Lexnames.

References

[1] Lesk. (1986) *Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone.* URL: https://dl.acm.org/citation.cfm?id=318728

[2] Agirre and Soroa (2009). *Personalizing PageRank for Word Sense Disambiguation.* URL: https://www.aclweb.org/anthology/E09-1005.pdf

[3] Raganato, Delli Bovi, and Navigli (2017). *Neural Sequence Learning Models for Word Sense Disambiguation* URL: https://www.aclweb.org/anthology/D17-1120

[4] Raganato, Collados, and Navigli (2017). *Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison.* URL: http://lcl.uniroma1.it/wsdeval/data/EACL17_WSD_EvaluationFramework.pdf