8 Text Chunking

Text chunking is another basic NLP task and consists in identifying segments, or *chunks*, of words that are syntactically related in a given sentence. Additionally, each chunk needs to be classified among some classes of interest, which gives the chunk type. In Figure 8.1, we present a sentence along with the corresponding chunking output. In this example, there are three chunk types: nominal (NP), verbal (VP), and prepositional (PP). Many text chunking systems cast this task as a sequence labeling problem by employing some appropriate tagging style. For instance, the row labeled IOB2 Tag in the figure indicates the token tags that encode the given chunks according to the IOB2 tagging style. This tagging style makes use of three prefix tags: B, for beginning, indicates the first token of a chunk; I, for inside, indicates any other token in a chunk; and O, for outside, indicates tokens that are not part of a chunk (most punctuation marks, for instance). Additionally, the B and I tags are combined with the chunk type.

Sentence	He	reckons	the deficit	will narrow	to	1.8 billion
Chunk	⎵	⎵	⎵	⎵	⎵	⎵
Type	nominal	verbal	nominal	verbal	prep.	nominal
IOB2 Tag	B-NP	B-VP	B-NP I-NP	B-VP I-VP	B-PP	B-NP I-NP

Figure 8.1: Text chunking example.

8.1 Task Formalization

We employ the IOB2 tagging style to cast text chunking as a sequence labeling problem. In that way, we use same modeling presented in the previous chapter, in which the input $\boldsymbol{x} = (x_1, \ldots, x_N)$ is a sequence of tokens and the output $\boldsymbol{y} = (y_1, \ldots, y_N)$ comprises a tag sequence, where $y_i \in S$ is the tag given for token x_i, and S is the IOB2 tagset.

8.2 Feature Factorization

Since we model text chunking as a sequence labeling problem, we use the observation and transition features and decompose them just as for the POS task, that is,

$$\mathbf{\Phi}(\boldsymbol{x},\boldsymbol{y}) = \left(\mathbf{\Phi}^{\text{obs}}(\boldsymbol{x},\boldsymbol{y}), \mathbf{\Phi}^{\text{trans}}(\boldsymbol{x},\boldsymbol{y})\right).$$

8.3 Prediction Problem

The prediction problem is exactly the same as presented in Section 7.3, that is, a longest path problem on an weighted directed acyclic graph. We also use the same loss function $\ell(\boldsymbol{y},\boldsymbol{y}') = \sum_{t=1}^{T} \mathbf{1}[y_t \neq y'_t]$, which counts the number of mislabeled tokens.

8.4 Basic Features

Our chunking datasets include the following basic features:

- *Word* – the word of each token;
- *POS tag* – the part-of-speech tag;
- *Basic form* – which type of word among the following types: (i) number; (ii) alphabetical in lower case; (iii) alphabetical in upper case; (iv) capitalized alphabetical; or (v) something else.

The basic features of each token also include the values of the aforementioned features for the three previous tokens and the three next tokens.

8.5 Empirical Results

We evaluate ESL on two text chunking datasets: the Bosque dataset (Fernandes et al., 2010b), a Portuguese language corpus; and the CoNLL-2000 dataset (Sang and Buchholz, 2000), an English language corpus. In Table 8.1, we present the number of chunks, sentences and tokens in each dataset. Both datasets have relatively the same size and are split into training and

Dataset	**Language**	**Training**			**Test**		
		Cks	**Sents**	**Tkns**	**Cks**	**Sents**	**Tkns**
Bosque	Portuguese	116,233	9,368	226,758	18,908	1,405	35,256
CoNLL-2000	English	106,978	8,936	211,727	23,852	2,012	47,377

Table 8.1: Basic statistics of the text chunking datasets.

test partitions. Performance for text chunking is reported on precision, recall and F-score. Precision is the percentage of recovered chunks that are correct and recall is the percentage of correct chunks that are recovered. A chunk is considered correct when both its span and its type are correct. The F-score is the harmonic mean of precision and recall. That is, $F = 2 \cdot P \cdot R/(P + R)$, where P is the precision ratio and R is the recall ratio.

8.5.1 Bosque Dataset

Fernandes et al. (2010b) propose a heuristic to extract text chunks from the Bosque treebank (Freitas et al., 2008). The Bosque corpus includes news articles comprising Brazilian and European Portuguese. Fernandes et al. also propose an ETL-based system for text chunking and evaluate it on the Bosque dataset. We report in Table 8.2 the performance of this system along with the performance of our ESL-based system. ESL reduces the error of the ETL system by 2.1%, regarding F-score. On the other hand, ETL achieves a higher precision than ESL.

System	P	R	*F*
ETL	**89.61**	85.41	87.46
ESL	88.06	**87.39**	**87.72**

Table 8.2: Performances on the Bosque dataset.

We use the following values for the ESL meta-parameters in order to train this system. The number of epochs is 20. The loss weight parameter C is set to 300. And, we generate feature templates containing from 2 to 4 features.

8.5.2 CoNLL-2000 Dataset

System	P	R	*F*
Wu et al. (2006)	**94.16**	**94.26**	**94.21**
Kudo and Matsumoto (2001)	93.47	93.49	93.48
ESL	94.05	94.18	94.12

Table 8.3: Performances on the CoNLL-2000 dataset.

The CoNLL-2000 Shared Task (Sang and Buchholz, 2000) provides an English text chunking dataset that includes some sections from the WSJ in the Penn Treebank. The best performing system on this dataset is presented by Wu et al. (2006). This system introduces a masking strategy to approach hard examples that involve words not seen in the training data. In Table 8.3, we present the performances of our ESL system, Wu et al.'s masking system, and the kernelized SVM presented in Kudo and Matsumoto (2001). The masking system outperforms ESL, achieving an error that is 1.5% smaller than ESL's. Nevertheless, ESL is still competitive to state-of-the-art systems.

By using the standard validation set, we select the following values for the ESL meta-parameters. The number of epochs is 50, the loss weight parameter C is set to 300, and all feature templates contain 2 features.