9
Quotation Extraction

Quotation extraction is to identify quotes and their authors in a given document. A quote is a segment of the input document and quotes cannot overlap each other. We consider that an input for this task, in addition to the document itself, also includes two sets: the candidate authors and the candidate quotes. Candidate quotes in the input can overlap. In that way, a feasible output for this task is a subset of non-overlapping quotes and their associated authors. In Figure 9.1, we exemplify this task. In this figure, authors are highlighted in bold type and quotes in italic. The subscripted numbers indicate the association between quotes and their respective authors. For instance, the author for the quote 'estranha' is Nélio Machado.

> **Nélio Machado**$_1$, que defende **Daniel Dantas**$_2$, considerou '*estranha*'$_1$ a acusação de que **Dantas**$_2$ teria cogitado subornar **o juiz**$_3$. '*Isso é o fim da picada. Completamente sem fundamento e bem no dia em que o* **Daniel**$_2$ *vai prestar depoimento. Estou inclinado a pedir suspeição* **dele**$_3$ *[***Fausto de Sanctis**$_3$*]. Acho muito estranho, tem conteúdo de mais armação do que qualquer outra coisa'*$_1$ disse **ele**$_1$.

Figure 9.1: Quotation extraction example.

9.1
Task Formalization

An input-output pair $(\boldsymbol{x}, \boldsymbol{y})$ for quotation extraction is represented as follows. The input $\boldsymbol{x} = (\boldsymbol{a}, \boldsymbol{q})$ comprises two sets: the candidate authors $\boldsymbol{a} = \{a_1, \ldots, a_K\}$ and the candidate quotes $\boldsymbol{q} = \{q_1, \ldots, q_N\}$. Each quote $q_i = (s_i, e_i)$, for $i \in \{1, \ldots, N\}$, is a segment in the document and is represented by its starting token s_i and its end token e_i, where $s_i \leq e_i$. The output $\boldsymbol{y} = (y_1, \ldots, y_N)$ is a vector of author indexes, where $y_i \in \{1, \ldots, K\} \cup \{0\}$ indicates the author associated to the quote q_i; and $y_i = 0$ means that q_i is not included in the output $\boldsymbol{y}$.

9.2
Feature Factorization

Fernandes (2012) proposes a structure learning modeling for quotation extraction that is based on an input feature vector $\boldsymbol{\Phi}(i,j) = (\phi_1(i,j), \ldots, \phi_M(i,j))$ that describes the candidate quote-author association (q_i, a_j). Then, for a given output $\boldsymbol{y}$, the global feature *vector* is defined as

$$\boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1,\ldots,N; y_i \neq 0} \boldsymbol{\Phi}(i, y_i),$$

which is the histogram of the local features for all associations selected in $\boldsymbol{y}$.

9.3
Prediction Problem

Here, the prediction problem is to find non-overlapping quotes associated to authors whose association weights are maximum. This problem can be reduced to the weighted interval scheduling (WIS) problem for which there is a known efficient dynamic programming algorithm. In order generate a WIS instance from a quotation extraction input $\boldsymbol{x}$, we create an weighted interval for each association (q_i, a_j). The segment, or span, for this interval is given by the quote segment (s_i, e_i); and, given the current model $\boldsymbol{w}$, the interval weight is

$$s(i,j) = \langle \boldsymbol{w}, \boldsymbol{\Phi}(i,j) \rangle.$$

Since the associations $(q_i, a_1), \ldots, (q_i, a_K)$ have the same span in the WIS instance – which is (s_i, e_i) – the WIS algorithm never selects more than one author for q_i. Additionally, if $s(i,j) < 0$ for all $j \in \{1, \ldots, K\}$, then q_i is not selected. And, clearly, overlapping quotes are never selected together. The weight of a complete solution $\boldsymbol{y}$ is then given by

$$\begin{aligned} s(\boldsymbol{y}) &= \sum_{i=1,\ldots,N; y_i \neq 0} s(i, y_i) \\ &= \sum_{i=1,\ldots,N; y_i \neq 0} \langle \boldsymbol{w}, \boldsymbol{\Phi}(i, y_i) \rangle \\ &= \langle \boldsymbol{w}, \boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{y}) \rangle. \end{aligned}$$

Thus, this WIS problem is equivalent to the prediction problem in the ESL framework.

We use a loss function that counts how many quotes have been associated to incorrect authors, that is $\ell(\boldsymbol{y}, \boldsymbol{y}') = \sum_{i=1,\ldots,N} \mathbf{1}[y_i \neq y'_i]$.

9.4 Basic Features

We use the same basic features from Fernandes (2012). The following basic features are used for each quote-author association (q_i, a_j):

- *Distance* – Number of tokens between q_i and a_j.
- *Direction* – Indicates whether a_j is on the left or on the right of q_i.
- *Say-verb between* – Indicates whether there is a *say-verb* between q_i and a_j. Fernandes proposes a list of say-verbs that are frequently used to indicate quotes, like say, speak and comment.
- *Number of say-verbs* – Number of say-verbs between q_i and a_j.
- *Author between* – Indicates whether there is an author between q_i and a_j.
- *Quote between* – Indicates whether there is a quote between q_i and a_j.
- *BLS* – Indicates whether Fernandes's baseline system selects (q_i, a_j) association.
- *Say-verb around* – Indicates whether a *say-verb* occurs at most two tokens away from q_i.
- *First letter uppercased* – Indicates whether the first letter in q_i is uppercased.

9.5 Empirical Results

We evaluate ESL on the GloboNotes dataset (Fernandes, 2012) that includes news articles from the Globo.com portal. These articles comprise ten different news genres, namely Sports, General, Celebrities, Arts, Economy, Education, Politics, Science, Technology, and World. This dataset is split into training and test subsets. In Table 9.1, we present basic statistic of these datasets.

	Docs	**Sentences**	**Tokens**	**Quotations**
Train	552	7,963	174,415	802
Test	133	1,834	41,613	205

Table 9.1: GloboNotes dataset statistics.

We compare ESL performance to ETL and a baseline system, both proposed by Fernandes (2012). We present these performances in Table 9.2. Performances are reported in precision, recall and F-score. We notice that ESL outperforms both ETL and the baseline system.

Model	Precision	Recall	*F*
ESL	**83.24**	71.49	**76.80**
ETL	69.44	**73.17**	71.26
Baseline	64.35	67.80	66.03

Table 9.2: Performances on the GloboQuotes dataset.

The results aforementioned are obtained by using the set of candidate authors that are manually annotated in the GloboNotes dataset. We follow the setting used in Fernandes (2012) to allow a fair comparison. On the other hand, the set of candidate quotes given to ESL are generated by simple rules that are part of Fernandes's baseline system. These rules select some segments from the document by applying a sequence of regular expressions. The extracted segments correspond to more than 90% of the quotes in the dataset. However, this heuristic still greatly reduces the number of segments given as input to the ESL system when compared to all possible segments.

Since GloboQuotes has no standard validation set, we use 5-fold cross-validation on the training set to tune ESL meta-parameters. The loss weight C is set to 10 and the number of epochs is 65. We generate templates containing from 2 to 4 basic features and, additionally, include templates by removing the feature at the root node of the decision tree.