NOVA

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

Prospeção e Análise de Dados Report

Prospeção e Análise de Dados
AY 2022/2023

João Padrão 58288
Riccardo Galarducci 66819

Contents

Abstract

The report presents performances and results of the implementation of *Relevant Expression LocalMaxs* and *Automatic Extraction of Implicit Keywords* algorithm.

1 Relevant Expressions LocalMaxs Extractor

We implemented *Relevant Expressions LocalMaxs Extractor* using the Python programming language. The algorithm extracts relevant expressions namely n-gram with $n \in [2, n]$ from a set of documents. The results presented in this section will consider n up to 7. We set the minimum frequency filter for an n-gram to be considered as a Relevant Expression equal to 2, so if a RE does not occur at least twice in the document will not be taken into consideration by the algorithm when selecting the relevant expressions. The p parameter, which corresponds to the exponent of the generalised mean, is set to 2.

Evaluation of the Algorithm Results. We tested the algorithm on an English corpus of 6.0Mw and a French corpus of 6.1Mw. For both, we used SCP and Dice cohesion functions. To evaluate its results we exploit *Precision Recall*, and *Fmetric* which are reported in Equation 1.

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN} \qquad Fmetric = \frac{TP_{Prc} + TP_{Rec}}{TP_{Prc} + TP_{Rec} + FP_{Prc} + FN_{Rec}} \tag{1}$$

where:

1. TP (*True Positives*) corresponds to the REs extracted by the algorithm which are relevant.
2. FP (*False Positives*) corresponds to the REs extracted by the algorithm which are not relevant.
3. FN (*False Negative*) corresponds to the REs not extracted by the algorithm which are instead relevant.

Precision. To compute the $Precision$ we have extracted 30 random REs from all the REs extracted by the algorithm. We then subjectively evaluated whether the relevant expressions were TP or FP. Finally, we compute the $Precision$.

Recall To compute the $Recall$ we needed the FN, so we extracted randomly some paragraphs from the corpus and, manually, look for REs inside those paragraphs until we found 20 REs. We then check over the collected REs if they were extracted as well from the $LocalMaxs$ algorithm (TP) or not (FN).

Fmetric The $Fmetric$ is the harmonic mean of $Precision$ and $Recall$, so we exploit the results obtained before to compute it. Table 1 and Table 2 summarize the evaluation metrics computed on the two corpus using SCP and $Dice$ respectively.

Table 1: Evaluation metrics with SCP_f glue function

SCP_f	EN6.0Mw	FR6.1Mw
$Precision$	0.76	0.7
$Recall$	0.8	0.8
$Fmetric$	0.78	0.74

Table 2: Evaluation metrics with $Dice\_f$ glue function

$Dice\_f$	EN6.0Mw	FR6.1Mw
$Precision$	0.73	0.66
$Recall$	0.8	0.65
$Fmetric$	0.76	0.66

The performances we obtained are higher using SCP_f as a cohesion function than $Dice_f$. When assessing the evaluation performance, we must take into account that results are subjective.

2 Automatic Extraction of Explicit and Implicit Keywords

The second task of the project consists in implementing an automatic extractor of explicit and implicit keywords from documents. The algorithm is computationally expansive, for this reason, the results shown below are performed on a corpus containing few documents.

2.1 Extract Most Informative REs.

We exploit our implementation of the LocalMaxs algorithm to extract from each document the relevant expressions. Then, in order to choose the ten most informative REs for each document we sort them in non-increasing order according to the **median length of the words** that constitute the REs and take for each document the first ten. The median length value can be used as a proxy of the informative capability of a sentence. We will consider these ten most informative REs as the explicit keywords of the documents.

2.2 Extract Informative Unigrams

Until now we only have the ten most informative multiword REs for each document. As the second step, we used **Tf-Idf** measure to extract the relevant unigram for each document. Tf-Idf returns a low value for not important words and a high value for relevant words. For this reason, we have defined a threshold parameter on Tf-Idf below which unigrams are not considered important and therefore discarded. We set the empirically, by analysing the Tf-Idf values, the threshold at 0.005.

2.3 Compute Semantic Proximity Between REs

As the last step, we compute the **Semantic Proximity** based on the **Inter-document Proximity** which assesses how correlated are the terms in all documents. Basically, if the REs (explicit keywords) of a document are correlated with relevant unigram or relevant multiword across the documents in the corpus then the latter will be considered as implicit keywords of the document under consideration. We do not include in Semantic Proximity the intra-document proximity component, this will result in a less precise extraction of implicit keywords. Table 3 shows some of the implicit keywords retrieved for a document which has *United States* as the explicit descriptor. The document under consideration does not have a defined theme. However, the explicit descriptor such as *Harward* or *Flag* provide interesting insights into the document.

Table 3: Implicit descriptor for document with *United States* Explicit descriptor

Corr Keyword	*Corr* Keyword	*Corr* Keyword
1.0 Harvard	1.0 flag	1.0 Newton's First Law
1.0 "macaroni products"	1.0 alternate jersey	1.0 maintaining law
1.0 Myanmar	1.0 Aung San	1.0 inertia