The underpinnings of a composite measure for automatic term extraction: the case of SRC

Carlos Periñán-Pascual

Abstract

The corpus-based identification of those lexical units which serve to describe a given specialized domain usually becomes a complex task, where an analysis oriented to the frequency of words and the likelihood of lexical associations is often ineffective. The goal of this article is to demonstrate that a user-adjustable composite metric such as SRC can accommodate to the diversity of domain-specific glossaries to be constructed from small- and medium-sized specialized corpora of non-structured texts. Unlike for most of the research in automatic term extraction, where single metrics are usually combined indiscriminately to produce the best results, SRC is grounded on the theoretical principles of salience, relevance and cohesion, which have been rationally implemented in the three components of this metric.

Keywords: automatic term extraction, salience, relevance, cohesion, SRC

1. Introduction

The extraction of both single- and multi-word terminological units from domain-specific corpora is fundamental not just for the construction of NLP resources (e.g. glossaries, thesauri, ontologies, etc.) but also as a stepping stone towards more complex computational tasks, such as information retrieval, text classification, document summarization or machine translation. The manual construction of specialized

glossaries is not only labor-intensive and time-consuming but also tends to be inconsistent, so automatic term extraction (ATE) systems can help solve this problem. In this context, DEXTER (Discovering and Extracting TERminology) was developed as an online multilingual ATE workbench[1] which is provided with a suite of tools to be used with non-structured text-based corpora. The contribution of DEXTER is to exploit advanced functionalities such as corpus compilation and management, document indexation and retrieval, query elaboration, textual exploration and terminological extraction within a framework where linguists, terminographers and knowledge engineers[2] can develop their research.

The goal of this article is to demonstrate the suitability of the metric devised for DEXTER, i.e. SRC (Salience, Relevance and Cohesion), from a theoretical and practical perspective. The remainder of this article is organized as follows: Section 2 briefly describes the state of the art in ATE; Section 3 thoroughly examines the three terminological features integrated in DEXTER, i.e. salience, relevance and cohesion; Section 4 explores the evaluation procedure, where two experiments are described and their results are discussed; and finally, Section 5 highlights the main conclusions.

2. An Overview of Automatic Term Extraction

The first ATE systems appeared with the advent of large electronic corpora of written texts in the 90s. Since the release of TERMINO (Plante and Dumas 1989) to the present, many projects have implemented different methods to identify terminological

[1] DEXTER, which has been developed in C# with ASP.NET 4.0, is intended to be freely accessible from the FunGramKB website (www.fungramkb.com). English, Spanish, French and Italian are currently supported in DEXTER.

[2] In fact, DEXTER enables data export to FunGramKB (Periñán-Pascual and Arcas-Túnez 2004; 2007; 2010; Mairal-Usón and Periñán-Pascual 2009; Periñán-Pascual and Mairal-Usón 2009; Periñán-Pascual 2013), a multipurpose lexico-conceptual knowledge base for natural language understanding systems.

units. In fact, there have been three main approaches to ATE, i.e. linguistic, statistic and hybrid, as described in the next paragraphs.

The linguistic approach to term recognition is typically performed by means of three consecutive tasks. First, words are tagged with their part-of-speech. Second, morphosyntactic patterns are used to capture acceptable surface realizations as term candidates. Third, a stop list of functional and generic words is applied. However, the linguistic approach to ATE involves two main problems. On the one hand, this model is remarkably restrictive, since those terms which do not match a linguistic pattern will never be captured. On the other hand, the process of building linguistic filters is language-dependent, so it becomes a time-consuming and labor-intensive task. Furthermore, the linguistic forms returned by linguistic filters cannot be defined as real terms. Indeed, pre-defined linguistic patterns serve as an indicator of *unithood*, instead of *termhood*, so there is a need to implement some model able to capture the notion of termhood after this linguistic stage.[3]

The statistical approach to term recognition is based on two different types of measures. On the one hand, lexical association measures such as X^2 (Nagao et al. 1976), *Pointwise Mutual Information* (Church and Hanks 1990), *T-score* (Church et al. 1991), *Dice coefficient* (Smadja 1993), *Log-Likelihood Ratio* (Dunning 1994) or *Jaccard similarity* (Grefenstette 1994), which serve to calculate the likelihood that two words can co-occur, have been frequently used in ATE. Indeed, although these unithood measures were not specifically devised for NLP, they have been employed for tasks

[3] Kageura and Umino (1996, 260-261) differentiated between unithood and termhood:

> 'Unithood' refers to the degree of strength or stability of syntagmatic combinations or collocations. (…) On the other hand, termhood refers to the degree that a linguistic unit is related to (or more straightforwardly, represents) domain-specific concepts.

such as collocation extraction, word sense disambiguation or the identification of translation equivalents.

Today it is common to find corpus management applications which apply this type of measure; for example, *Collins WordBanks Online* (2013) employs mutual information and T-score to measure statistical significance in Collins' corpora. On the other hand, there are also some statistical measures for termhood, such as *TF-IDF* (Singhal et al. 1996), *C-value* (Frantzi and Ananiadou 1996; Barrón-Cedeño et al. 2009), *Weirdness* (Ahmad et al. 2000), *Domain-Specificity* (Park et al. 2002), and *Domain-Pertinence* and *Domain-Consensus* (Sclano and Velardi 2007; Navigli and Velardi 2002), being most of them the major focus for my research, as discussed in the next section.

Finally, it should be noted that few ATE systems adopt a purely statistical approach, since "the direct application of sole statistical measures to not-linguistically-filtered expressions can lead to a terminology rich of unwished forms" (Pazienza et al. 2005, 259). Therefore, there is a general tendency to consider both linguistic and statistical properties for the identification and extraction of term candidates. One of the most popular examples of hybrid model is found in *C-value/NC-value* (Frantzi et al. 2000), where the NC-Value incorporates contextual information into the C-Value, which focuses on nested terms.

Linguistic, statistic and hybrid approaches have traditionally relied on the processing of a single target corpus, i.e. the specialized corpus. However, ATE can also be grounded on the contrastive analysis between corpora, more particularly, on the differences in the distribution of lexical items between the target corpus and a reference corpus, i.e. general-language corpus (cf. Ahmad et al. 2000; Peñas et al. 2001; Park et

al. 2002; Wong et al. 2007), or between the target corpus and other specialized corpora (cf. Sclano and Velardi 2007).

As noted by Conrado et al. (2014, 1), "although ATE has been researched for more than 20 years, there is still room for improvement." This article focuses on the issue that any ATE system should be provided with an adequate statistical measure. But what does "adequate" mean in this context? Since the effectiveness of probabilistic measures is closely dependent on the characteristics of the corpus, the statistical adequacy of the metric implies that the system should be able to cope with a wide diversity of corpus designs by outperforming the results of other metrics. In this regard, I intend to increase the precision, as well as reducing the silence, and distribute false positive candidates as effectively as possible, trying to place most of them at the bottom of the list of extracted candidates. To this end, my research is based on two basic assumptions: (i) the metric should be composite (i.e. a metric whose components are defined by other metrics), so there is no need to devise new metrics but to combine existing ones on a rational basis; and (ii) the metric should be user-adjustable (i.e. a metric which contains parameters whose values can be adjusted by the user), so that it can accommodate to the configuration of the document collection. These terminological issues are examined in detail in the next section.

3. Term Extraction Metric in Dexter

Although there are still some studies that focus on the engineering of new measures for unithood and termhood, the recent trend is to combine statistical features effectively (Fedorenko et al. 2013). Throughout this section, and to illustrate the statistical measures involved in DEXTER, I examine a sample collection consisting of the extracts

(2), (3) and (4), which are going to be treated as documents describing the domain of electronics for computer hardware. More realistic corpora are taken into account in Section 4, where the metric is evaluated.

(2) *A chip is an integrated circuit, which consists of transistors, capacitors and resistors. The transistor amplifies the electric current. The capacitor filters the electric current. The resistor reduces the flow of electric current.*

(3) *The transistors, resistors, and capacitors are installed on a foundation of silicon, which is a semiconductor.*

(4) *The benefits of miniaturization are that chips become smaller, faster and cheaper, but the problem is overheating.*

It is important to note that the smallest unit of analysis in this research is the stemmed ngram. My motivation of stemming is twofold. First, stemming can serve to convert various strings expressing the same meaning into a single form. For example, in Experiment 1 in Section 4, the same stemmed trigram was recognized for the linguistic realizations *light dependent resistor* and *light dependant resistor*. However, one of the consequences of this type of linguistic normalization is that recall usually becomes higher but precision is lower (Kraaij and Pohlmann 1996), i.e. the removal of surface variations often enables to find more similarities but at the expense of losing some semantic significance. This approach is particularly useful for the small- and medium-sized corpora which can be built with DEXTER. However, if a large corpus is managed, then precision can be prioritized and consequently stemming should not be involved. Second, I opted for a stemmer rather than just a lemmatizer because the former dramatically reduces the number of candidates to inspect during validation, besides the fact that it is more effective to determine terminological prominence on stems instead of focusing on lexical forms or lemmas.

The ngram in DEXTER is a sequence of *n* contiguous lexical tokens, where $0 < n \leq 3$. When the ngram is undergoing a validation process, I will call it "candidate". Indeed, only after the validation and lemmatization of the ngram, the candidate can be really called "term". Therefore, the term is a lexical unit (i.e. lexical variant or lexeme) which originated from an ngram. The traditional notion of term has been here adopted, that is, a lexical unit which corresponds to a conceptual unit, i.e. "terms are the linguistic representation of concepts" (Sager 1990, 57), which in turn is part of a given specialized-knowledge domain.

3.1. Term Salience

The task of automatic document indexation has a clear point in common with that of ATE, since the keywords employed to index a given document are usually perceived as terminological units (Pazienza et al. 2005). This idea is supported by the fact that indexation terms are words or sequences of words which have sufficient semantic load so as to provide information about the substance of the content in a document.[4] Therefore, one of the pillars of the DEXTER metric is the notion of salience, which is based on the termhood measure TF-IDF (Salton 1971; Salton and Yang 1973; Salton et al. 1975; Salton et al. 1975; Salton and Buckley 1988; Singhal et al. 1996), i.e. the weight of a term is determined by the relative frequency of the term in a certain document (or term frequency, i.e. TF) compared with the inverse proportion of that term in the entire document collection (or inverse document frequency, i.e. IDF). This measure has been frequently applied in the field of information retrieval, where the indexed documents in a repository (e.g. database) are ranked automatically on the basis

[4] Semiologically, Barthes (1964) defined the "substance of content" of the linguistic sign as that part which "includes, for instance, the emotional, ideological, or simply notional aspects of the signified, its 'positive' meaning".

of the keywords present in a given query. Indeed, TF-IDF is one of the most popular AKE (Automatic Keyword Extraction) measures, which can derive lists of keywords from a collection of documents and where each one of these keywords is typically assigned a weight representing how salient the keyword is to the selected document. Therefore, TF-IDF can be used to weight how well the words and sequences of words in a corpus describe the contents of the documents, in such a way that if the keyword is given a high value, then it is more related to the topic of the document than a keyword with a low value.

Thus, the TF-IDF of a stemmed ngram *g* can be calculated by applying different weighting schemes to the following formula:[5]

(5)
$$TF\text{-}IDF(g) = TF(g) * IDF(g) * NORM(g)$$

where *NORM* is a normalization factor. Some of the most popular alternative equations for *TF*, *IDF* and *NORM* are described as follows, where the SMART notation is placed on the right-hand side:[6]

(6a)	$TF(g) = f_d(g)$	n [natural]
(6b)	$TF(g) = 1 + \log(f_d(g))$	l [logarithm]
(7a)	$IDF(g) = 1$	n [none]
(7b)	$IDF(g) = 1 + \log\left(\frac{N_T}{df_T(g)}\right)$, where $df_T(g) > 0$	t [idf]
(8a)	$NORM(g) = 1$	n [none]
(8b)	$NORM(g) = \frac{1}{\sqrt{\sum_{g \in d} (TF(g) \times IDF(g))^2}}$	c [cosine]

[5] Sabbah and Abuzir (2005) demonstrated that results from the TF-IDF technique are improved when stemming and stop-words removal are applied.

[6] SMART (Salton 1971) is one of the first information retrieval systems based on the vector space model. Not all the notations of the SMART term-weighting schemata are consistent, since the system has been developed for over 40 years; in this article, I follow the notation suggested by Singhal, Salton, and Buckley (1996), which has become very popular thanks to Manning, Raghavan, and Schütze (2009).

where d is a document in the specialized corpus CP_T, N_T is the number of documents in CP_T, $f_d(g)$ is the number of occurrences of the ngram in d, and $df_T(g)$ is the number of documents in which the ngram appears in CP_T. The descriptions of the symbols used in all of the equations presented in this article are included in Appendix 1.

The rationale for the three components of the equation (5) is described as follows. First, *TF(g)* serves to justify the fact that more weight is given to those ngrams that appear many times in a given document. Second, *IDF(g)* rewards those ngrams which are concentrated just in a few documents of the corpus. Thus, the value of a rare ngram in the corpus is high, whereas the value of a frequent ngram is low; in other words, less weight is given to ngrams that appear in many documents. Third, the document size is a parameter which can dramatically affect the calculation of weights, since (i) long documents usually use the same ngrams repeatedly, and (ii) long documents have numerous different ngrams (Singhal et al. 1996). Therefore, *NORM(g)*, i.e. document length normalization of ngram weights, is used to remove the advantage of long documents: less weight is given to documents that contain many ngrams. In other words, the normalization factor makes all documents be treated equally important regardless of their size. Moreover, when the cosine normalization is employed, the TF-IDF value ranges from 0 to 1. This will make it easier to compare the weight of terms between corpora.

In DEXTER, the document weighting scheme for TF-IDF is *ntc*,[7] where the document vector has natural term frequency, idf and cosine normalization. This weighting scheme was actually chosen on the grounds that the effectiveness of a

[7] As explained in Singhal et al (1996, 153):

> In the Smart system, term weighting schemes are denoted by triples of letters. The first letter in a triple is a short hand for the function of the term frequency factor being used in the term weights, the second letter corresponds to the inverse document frequency function and the third letter corresponds to the normalization factor applied to the term weights.

probabilistic measure is closely dependent on the characteristics of the document collection. Following some of the conclusions derived from experimental research (cf. Salton and Buckley 1988, 521; Singhal 1997, 9), here are some guidelines for choosing the most effective setting of the TF-IDF weighting scheme:

(a) For the *TF(g)* component, function *l* is typically used for large full-text collections, and function *n* for other cases.
(b) For the *IDF(g)* component, function *n* (i.e. no *idf*) is typically used for dynamic document collections with many changes in the collection makeup, and function *t* for other cases.
(c) For the *NORM(g)* component, function *n* (i.e. no normalization) is typically used for short documents of homogeneous length, and function *c* for other cases.

DEXTER has been devised to build small- and medium-sized corpora, which are relatively static and whose documents can be of variable length, so I conclude that the most appropriate weighting scheme should be *ntc*. Therefore, the salience of the ngram *g* in the document *d* is calculated with the following formula, which results from combining equations (6a), (7b) and (8b):

(9)

$$S_d(g) = \frac{f_d(g) \times (1 + \log_2 \left(\frac{N_T}{df_T(g)}\right))}{\sqrt{\sum_{g \in d} \left(f_d(g) \times \ (1 + \log_2 \left(\frac{N_T}{df_T(g)}\right))\right)^2}}$$

It should be noted that the cosine normalization (8b) is calculated in DEXTER on the basis of the type of ngram, i.e. unigram, bigram or trigram. For example, the weight of a

certain bigram in a given document is normalized by calculating the weights of all and only the bigrams in the same document.

But how should the $S_d(g)$ value be interpreted? To answer this question, you need to understand that a corpus in DEXTER is perceived as a "vector space model" (VSM). The principle behind the VSM is to use frequencies in a text corpus as a clue to discover semantic information.[8] Thus, DEXTER allows every document in the corpus to be represented as a vector (or dot) in a common vector space:

$$\delta_{i,T} = \left(S_{d_i}(g_1), S_{d_i}(g_2), S_{d_i}(g_3), \ldots, S_{d_i}(g_{k-1}), S_{d_i}(g_k)\right) \tag{10}$$

where k is the total number of ngrams in document d. Therefore, $\delta_{i,T}$ is the vector representation of the i-th document in CP_T. For example, if $\delta_{4,T}$ and $\delta_{7,T}$ are perceived to be semantically similar, then the two dots are expected to be closer in the space. The representation (10) is mathematically known as a bag, since the exact order of the items is not relevant. Thus, vectors do not capture the structure of phrases in sentences, so the sequential order of words is lost. Nevertheless, "in spite of this crudeness, (…) vectors seem to capture an important aspect of semantics" (Turney and Pantel 2010, 147), since "it seems intuitive that two documents with similar bag of words representations are similar in content" (Manning et al. 2009, 117). In fact, this widely-held view in NLP has a close connection with the distributional hypothesis, which states that words occurring in similar contexts tend to have similar meanings (Harris 1954).

[8] VSMs were originally used in information retrieval, but they have currently inspired many researchers to extend them to other tasks in NLP, e.g. document classification, essay grading, thesaurus generation, or word sense disambiguation, among many others. See Turney and Pantel (2010) for a detailed survey of the applications of the VSM.

For the sake of greater precision, it should be said that DEXTER is provided with three separate vectors for a single document, i.e. one for each type of ngram, since the weight of an ngram in a given document is normalized with respect to all and only the ngrams of the same type in that document. Assuming that:

> UG_T is the set of unique unigrams in CP_T, so $UG_T(d_i)$ represents all the unigrams in the *i*-th document,
> BG_T is the set of unique bigrams in CP_T, so $BG_T(d_i)$ represents all the bigrams in the *i*-th document, and
> TG_T is the set of unique trigrams in CP_T, so $TG_T(d_i)$ represents all the trigrams in the *i*-th document,

the vector representation of all the unigrams, bigrams and trigrams for the *i*-th document in CP_T would be represented respectively as follows:

(11) $\delta_{i,UG_T} = \left(S_{d_i}(g_1), S_{d_i}(g_2), S_{d_i}(g_3), \dots, S_{d_i}(g_{|UG_T|-1}), S_{d_i}(g_{|UG_T|})\right)$, where $g_n \in UG_T(d_i)$

(12) $\delta_{i,BG_T} = \left(S_{d_i}(g_1), S_{d_i}(g_2), S_{d_i}(g_3), \dots, S_{d_i}(g_{|BG_T|-1}), S_{d_i}(g_{|BG_T|})\right)$, where $g_n \in BG_T(d_i)$

(13) $\delta_{i,TG_T} = \left(S_{d_i}(g_1), S_{d_i}(g_2), S_{d_i}(g_3), \dots, S_{d_i}(g_{|TG_T|-1}), S_{d_i}(g_{|TG_T|})\right)$, where $g_n \in TG_T(d_i)$

From this approach, the whole corpus in DEXTER would be represented as three ngram-document matrices, one for each type of ngram. For example, the unigram-document matrix for our three-document collection has $|UG_T|$ rows, i.e. one for each g in $UG_T(d_i)$, and N_T columns, i.e. one for each d in CP_T. In particular, $|UG_T| = 25$ and $N_T = 3$. Each cell in this matrix represents $S_{d_i}(g_j)$, where $j \leq |UG_T|$ and $i \leq N_T$, and if $g_j \notin d_i$, then $S_{d_i}(g_j) = 0$. Thus, $UG_T(d_i)$ contains all those unigrams in the first document (i.e. column d_1) whose weight is higher than zero.[9]

[9] Most of the documents generally employ a small fraction of the whole vocabulary, so ngram-document matrices usually become very sparse, that is, the weight assigned to most of the elements is zero. However, as noted by Turney and Pantel (2010), sparsity can be seen just as a temporary problem of lack

To illustrate, in the document (2) of our collection the weight of the stemmed unigram *capacit*, corresponding to the token *capacitor*, would be calculated as (13).

(14) $$S_{d_1}(g_4) = \frac{2 \times 1.58496}{\sqrt{1.58496^2 + 2.58496^2 + \cdots + 7.75488^2}} = \frac{3.16992}{14.13181} = 0.22431$$

Here the unnormalized weights (i.e. *TF(g) x IDF(g)*) of the first unigram (i.e. *chip*) and the last unigram (i.e. *current*) in this document are 1.58496 and 7.75488 respectively.

Recall that salience is based on TF-IDF, so $S_d(g)$ indicates how unique a term is within a document. For example, in $UG_T(d_1)$ the weights of unigrams such as *flow* and *current* are 0.18291 and 0.54875 respectively. This means that *capacitor*, whose weight is 0.22431, is more salient than *flow* but less than *current* for the content of the first document. This salience is calculated on the basis of the word stem, resulting more effective than taking the word form or the lexeme as the unit of analysis. Thus, the semantic burden of words such as *capacitor*, *capacitors* and *capacitance* is jointly measured.

But how can the salience of ngrams be calculated with respect to the whole collection and not just to a single document? The salience of the ngram *g* in the whole corpus CP_T is calculated as the normalized average of the weights of *g* in CP_T:

(15)

$$S_T(g) = \frac{\sum_{d \in CP_T} S_d(g)}{\sqrt{\sum_{g_k \in CP_T} (S_T(g_k))^2}}$$

---

of data: the more documents a collection has, and the larger the documents are, the fewer zeroes the matrix will have.

Again, the normalization factor of this formula only takes into account ngrams of the same type. Thus, the salience index of a certain unigram in a corpus is normalized by calculating the average of the weights of all and only the unigrams in the same corpus. Therefore, the vector representation of the unigrams in CP_T would be represented as follows:

(16) $$\Delta_{UG_T} = (S_T(g_1), S_T(g_2), S_T(g_3), \dots, S_T(g_{k-1}), S_T(g_k)), \text{ where } g \in UG_T$$

In the example above, the averaged weight of the unigram *capacit* in our three-document collection would be calculated as (17).

(17) $$S_T(g_4) = \frac{0.22431 + 0.27076 + 0}{\sqrt{0.09903^2 + 0.18706^2 + \cdots + 0.26802^2}} = \frac{0.49507}{1.84713} = 0.26802$$

We can see that $S_T(g_1)$ (i.e. for the unigram *amplifi*) and $S_T(g_k)$ (i.e. for the unigram *transist*) in the collection are 0.09903 and 0.26802 respectively. For example, the averaged weights of the unigrams *flow* and *current* are 0.09903 and 0.29708 respectively, so we can infer that *capacitor* is more salient than *flow* but slightly less than *current* for the content of the collection. Although $S_d(g)$ is employed as the basis for $S_T(g)$, the ranking of ngrams in a given document cannot be extrapolated to the whole corpus. In fact, *instal* is more salient than *capacit* in the second document but it isn't with respect to the whole collection. To end this section, it can be definitely concluded that the salience index becomes a powerful termhood measure.

3.2. Term Relevance

Another issue about termhood, however, is the difference between prevalence and tendency, as explained by Wong et al. (2008). Salience measures the prevalence of the term in a particular target domain, but it does not reflect the tendency of term usage across different domains. In other words, TF-IDF fails to comprehend that terms are also properties of domains, and not just of documents. For example, in the previous document collection, it can be stated that *capacitor* (i.e. $S_T(capacit)$ = 0.26802) is more salient than *flow* (i.e. $S_T(flow)$ = 0.09903). This implies that *capacitor* has high prevalence within that selection of documents which describe the domain of electronics for computer hardware. However, it is not possible to assure that this relation of prevalence will remain the same with respect to another collection of documents describing the same domain. Consequently, I propose to combine the salience of TF-IDF with a corpus-based termhood measure, quantifying the relevance of ngrams through the contrastive analysis between the target corpus and a reference corpus. It is important to keep in mind that there is a statistical feature behind every measure applied in DEXTER, in such a way that the combination of all the features enables us to get a more precise weight for the term. Indeed, a critical choice in this type of combinatorial method is the selection of features, where the criteria of redundancy and irrelevance should be taken into account (Fedorenko et al. 2013, 19):

> Having a lot of different features, the goal is to exclude redundant and irrelevant ones from the feature set. *Redundant* features provide no useful information as compared with the current feature set, while *irrelevant* features do not provide information in any context.

The remainder of this section describes the reasons which led me to support or rule out some metrics oriented to measure the relevance of ngrams.

One of the most influential studies on termhood measures based on contrastive corpora has been Ahmad et al. (2000), who proposed the index of the weirdness of specialised terms.[10] By adapting this metric to our unit of analysis, i.e. stemmed ngrams, we could quantify the relevance of a given *g* in CP_T as follows:

(18)

$$R_T(g) = \frac{P_T(g)}{P_R(g)}$$

$$P_T(g) = \frac{f_T(g)}{|CP_T|}$$

$$P_R(g) = \frac{f_R(g)}{|CP_R|}$$

where $P_T(g)$ is the probability of the ngram in CP_T and $P_R(g)$ is the probability of the ngram in CP_R, so $f_T(g)$ is the frequency of the ngram in CP_T, where $|CP_T|$ is the total number of words in the target corpus, and $f_R(g)$ is the frequency of the ngram in CP_R, where $|CP_R|$ is the total number of words in the reference corpus. In this setting, if an ngram is used more frequently in D_T than in D_R, then the relevance index of the ngram is greater than 1, and conversely. It should also be noticed that if the ngram does not occur in CP_R, then $f_R(g) = 1$.

[10] For example, the equation of Domain Specificity (Park et al. 2002) is very similar to that of weirdness.

Some other corpus-based metrics were grounded on the comparative analysis across different specialized domains, e.g. Velardi et al.'s Domain Relevance (2001). However, they were not taken into account in my research, since I intended to minimize the linguistic resources provided to DEXTER, a crucial issue to facilitate the scalability of this multilingual ATE system.

On the other hand, the metric of weirdness was extended to the index of relevance by Peñas et al. (2001) as follows:

(19)

$$R_T(g) = 1 - \frac{1}{log_2(2 + \frac{P_T(g) \times df_T(g)}{P_R(g)})}$$

where $df_T(g)$ is the document frequency of g in CP_T. Apart from normalizing the term weight, this new metric simply integrates the document frequency to the weirdness formula. Although there has been some experimental evaluation of the performance of TF-IDF together with the measure (19) (cf. Fedorenko et al. 2013), I conclude that these two statistical features become partially incompatible. You should recall that in TF-IDF ngrams which are concentrated just in a few documents of the corpus are considered to be more salient (i.e. inverse document frequency). In this measure of relevance, however, ngrams which appear in a very small fraction of the documents in the collection are considered to be less relevant (i.e. document frequency). Our strategy has been to remove the document frequency component from the previous equation, which is basically the same as normalizing the weirdness index:

(20)

$$R_T(g) = 1 - \frac{1}{\log_2\left(2 + \frac{P_T(g)}{P_R(g)}\right)}, \text{iff } |g| = 1$$

where $|g|$ is the number of lexical items included in the ngram.

It should also be recalled that the weirdness metric was originally devised for unigrams. In case of multi-word terms, I could have employed the frequencies of the bigrams and trigrams in CP_R, but in a multilingual workbench such as DEXTER this requirement would have negatively impacted on the scalability of the system. Consequently, and like Knoth et al. (2009), I chose to calculate the relevance of complex candidates in CP_T on the basis of the geometric mean of each lexical item within the candidate, that is:

(21)

$$P_T(g) = \frac{\sqrt[|g|]{\prod_{k_i \in g} f_T(k_i)}}{|CP_T|}, \text{iff } |g| > 1$$

$$P_R(g) = \frac{\sqrt[|g|]{\prod_{k_i \in g} f_R(k_i)}}{|CP_R|}, \text{iff } |g| > 1$$

where $f_T(k)$ and $f_R(k)$ represent the frequency of a given unigram in g with respect to CP_T and CP_R respectively. With the geometric mean, I can minimize the effects of extremely small or large values in a skewed frequency distribution of the items within the multi-word candidate. Moreover, this approach does not require you to build or

search a whole corpus of reference for every language, but only to have the frequency list of the tokens in the general-language corpus. The current version of DEXTER uses the *British National Corpus* (BNC)[11] and *Corpus de Referencia del Español Actual* (CREA)[12] as the corpora of reference for English and Spanish respectively. The lexical inventories of both corpora actually required some pre-processing. First, those tokens containing a digit or any other non-alphabetical character were removed. Second, the remaining tokens were stemmed, so we added the frequencies of all those tokens whose stems were the same. After applying these two pre-processing techniques, the lexical inventory of the BNC was eventually reduced to 25% and that of CREA to 55%. Finally, those tokens whose frequency was 1 were removed; this amounts to 40% of the stems in the case of the BNC, and to 45% in the case of CREA. Thus, the frequency of any ngram which is not found in the list will finally be 1, avoiding the problem which arises when any of the frequency values in the dividend of the metric (21) is 0, where the geometric mean would have been 0. Therefore, when *g* does not occur in CP_R, then $f_R(g) = 1$, whose normalized frequency is 1×10^{-8} in the BNC and 6.6×10^{-9} in CREA.

Returning to our document collection, it can now be asserted that the degree of relevance of the ngram *capacit* with respect to CP_T is 0.93023, being calculated as:

(22)

$$R_T(\text{capacit}) = 1 - \frac{1}{\log_2\left(2 + \frac{0.06}{0.000002907}\right)} = 0.93023$$

[11] The unlemmatised frequency list of English words was downloaded on 23 June 2014 from http://www.kilgarriff.co.uk/bnc-readme.html.

[12] The unlemmatised frequency list of Spanish words was downloaded on 23 June 2014 from http://corpus.rae.es/lfrecuencias.html.

Since $R_T(amplifi) = 0.90955$ and $R_T(flow) = 0.87381$, I conclude that both *capacitor* and *amplifier* are more relevant than *flow*; however, it should be recalled that *amplifier* is as salient as *flow*.

3.3. Term Cohesion

In the case of bigrams and trigrams, I also introduced the notion of cohesion, serving to determine the unithood of complex ngrams. In the same vein as current ATE systems, DEXTER adopts a hybrid approach where termhood and unithood are combined to produce a unified weight. As Zhang et al. (2008, 2112) demonstrated, "hybrid methods work better than 'termhood' only methods". Therefore, whereas salience and relevance serve to measure termhood, cohesion is aimed to quantify the degree of stability of multi-word terms. Although many standard association measures for unithood have been proposed in other works (cf. Section 2), Park et al. (2002, 5) reminded us that these measures have two major drawbacks:

> First, they evaluate the degree of association between two units and need to apply special techniques to calculate the association of terms with more than two words (…). Second, these measures tend to give higher values for low frequency terms, especially mutual information.

Moreover, Korkontzelos et al. (2008, 249) showed that:

> Termhood-based approaches which take into consideration the nestedness of a candidate term into others (…) have in general superior performance over methods which measure the strength of association among the tokens of a multi-word candidate term.

In this regard, one of our first options was to consider C-value (Frantzi and Ananiadou 1996; Frantzi et al. 2000), which is described as "a method to improve the extraction of nested terms" (Frantzi et al. 2000, 122). From the perspective of combining several statistical features, it is important to keep in mind that this metric serves to measure termhood as well as unithood, i.e. "C-value measures 'termhood' by using term frequencies, and 'unithood' by examining frequencies of a term used as parts of longer terms" (Zhang et al. 2008, 2109). In fact, both components are clearly separated in the C-value formula, where unithood resides in the NST part and termhood in the remainder of the following equation:

(23)

$$CV(g) = \log_2|g| * f_T(g), \text{ iff } |S_g| = 0$$

$$CV(g) = \log_2|g| * (f_T(g) - NST(g)), \text{ iff } |S_g| > 0$$

$$NST(g) = \frac{\sum_{b \in S_g} f_T(b)}{|H_g|}$$

where $|g|$ is the number of unigrams within g, being $|g| > 1$, H_g represents the set of distinct complex ngrams that contain g and $|H_g|$ is the number of ngrams in H_g. In case that g does not appear as nested, C-value assigns a value based on the number of tokens of the candidate and its frequency of occurrence in CP_T. It should be noted that $|g|$ is a key factor: a longer ngram appearing n times in a corpus is more important than a

shorter ngram appearing *n* times in the same corpus, since it is less probable that the longer ngram will occur more frequently than the shorter one.

When *g* is found as part of any other longer ngram, then the last formula in (23) serves to measure the nestedness of *g*. With respect to nestedness, the goal of C-value is not only to "avoid the extraction of substrings that are not terms", but also to "extract those substrings that are terms" (Frantzi et al. 2000, 117). For example, in *floating point constant*, *floating point* is a term but this is not the case of *point constant*. Nestedness is here quantified as the degree of independence of a complex ngram; in other words, the greater the number of longer ngrams in which *g* appears as nested, the smaller the independence of *g*.

There are two features of this method that attracted my attention. First, in the case of bigrams, C-value is proportional only to the frequency, since $\log_2(2) = 1$. This implies some serious consequences for small- or medium-sized corpora. For example, in our sample collection, the only complex terms which can be found are *integrated circuit* and *electric current*. Here their C-values are equal to their corresponding frequencies, i.e. 1 and 3 respectively. On the other hand, the candidate *capacitor filters*, which cannot be really considered a term, is also assigned 1. In all these cases, $\log_2(|g|) = 1$ and the NST component is not taken into account because $|H_g| = 0$. Therefore, in case of bigrams not being nested in longer ngrams, can we really assure that this metric serves to measure unithood? Second, only when the *NST* value is equal to $f_T(g)$, then C-value = 0. However, the interpretation of this zero value can be misleading on the premise that the higher the C-value, the higher the probability that a multi-word candidate can become an independent term. To illustrate, suppose that the bigram *control circuit* has appeared in only one longer candidate term (e.g. *motor control circuit*):

(24)

$$f_T(motor\ control\ circuit) = 4$$

$$f_T(control\ circuit) = 4$$

$$CV(control\ circuit) = 4 - \frac{4}{1} = 0$$

and *PC power* is nested within several other candidates, as shown in (25):

(25)

> *The standard PC power supply unit has two safety mechanisms that prevent it from being switched "ON" without the motherboard attached.*
> *[…]*
> *An old PC power supply unit makes an excellent and cheap bench top power supply for the electronics constructor.*[13]

$$f_T(standard\ PC\ power) = 1$$

$$f_T(PC\ power\ supply) = 2$$

$$f_T(old\ PC\ power) = 1$$

$$f_T(PC\ power) = 2$$

$$CV(PC\ power) = 2 - \frac{4}{3} = 0.67$$

According to the C-value of *control circuit* and *PC power*, I could wrongly conclude that the latter is more independent than the former. However, this assumption does not provide a true view of reality. Whereas *control circuit* has become a stable term in the field of electronics, as shown in the following lexicographical entry:

> **control-** Also called a control circuit. 1. In a digital computer, those parts that carry out the instructions in proper sequence, interpret each instruction, and apply the proper signals to the arithmetic unit and other parts in accordance with the interpretation. (Graf 1999)

[13] Text extracted from http://www.electronics-tutorials.ws/blog/convert-atx-psu-to-bench-supply.html.

PC power is more frequently found as part of compounds such as *PC power consumption, PC power speed* or *PC power supply* in the computer hardware domain.

Therefore, these two features clearly demonstrate that C-value is not always an adequate method for any type of document collection. In fact, most of the research with C-value has focused on long candidate terms in the medical domain, where 4-gram and even 5-gram candidates are relatively frequent, e.g. *adenoid cystic basal cell carcimona* (cf. Frantzi et al. 2000). However, although it is usually agreed that there is a significant proportion of terminological noun phrases in specialized corpora, not all domains are described in the same degree of linguistic complexity. For example, Golik et al. (2013, 159-160) concluded that, after the exploration of biomedical corpora as well as the consultation of domain experts, 5-grams structured linguistically as "*NN in NN, NN for NN, NN at NN* and *NN to NN* are interesting and useful to consider", but they also acknowledged that these patterns are certainly infrequent in other specialized domains.

In the search for other unithood methods away from standard association measures, I also found Term Cohesion (Park et al. 2002), a metric which computes the degree of cohesion among the items that compose a multi-word candidate term. Again, taking the ngram as the unit of analysis, the Term Cohesion of a complex ngram (*g*) can be estimated as follows:

(26)

$$C_T(g) = \frac{|g| \times f_T(g) \times \log_2(f_T(g))}{\sum_{k_i \in g} f_T(k_i)}, \text{iff } |g| > 1$$

where $f_T(g)$ is the frequency of *g* in CP_T and $f_T(k)$ is the frequency of a given unigram in *g* with respect to CP_T. Like C-value, Term Cohesion is proportional to the length and the frequency of the complex ngram. However, Term Cohesion does not take into

account the number of different candidates in which *g* appears as nested, but only the frequency of the items which compose *g*. More particularly, cohesion is high when the items which compose the ngram are more frequently found within the ngram than alone in texts. In this line, I propose that the logarithmic component should not be included when $f_T(g) = 1$; otherwise, regardless of the values in the other components of this measure, cohesion will always be 0 in these cases. Therefore, cohesion can be calculated more accurately as follows, where *F* is the logarithmic factor:

(27)

$$C_T(g) = \frac{|g| \times f_T(g)}{\sum_{k_i \in g} f_T(k_i)} \times F, \text{ iff } |g| > 1$$

$$F = \begin{cases} 1, & \text{iff } f_T(g) = 1 \\ \log_2(f_T(g)), & \text{iff } f_T(g) > 1 \end{cases}$$

Indeed, we can realize that the first component of this equation estimates a value which can also be calculated as the frequency of the ngram over the arithmetic mean of the frequency values of their nested unigrams, so:

(28)

$$C_T(g) = \frac{f_T(g)}{\left(\frac{\sum_{k_i \in g} f_T(k_i)}{|g|}\right)} \times F$$

In this context, I chose to replace the arithmetic mean by a geometric mean. This minor change is motivated by two facts. On the one hand, since all frequency values are positive numbers, geometric mean is smaller than arithmetic mean, so cohesion scores will be higher. However, geometric mean remains exactly the same as arithmetic mean

when all the frequency values involved in the complex candidate are equal, e.g. in the case that nested unigrams do not appear alone in CP_T. On the other hand, and similarly to the relevance metric, geometric mean smoothes the result in a frequency distribution where extreme values are present. For example, in the case of trigrams of the type "N prep N", the high frequency of prepositions makes distribution values be skewed right, resulting in a median which is closer to the geometric mean than the arithmetic mean. Consequently, cohesion is finally calculated as follows:

(29)

$$C_T(g) = \frac{f_T(g)}{\sqrt[|g|]{\prod_{k_i \in g} f_T(k_i)}} \times F, \text{iff } |g| > 1$$

Returning to our example, the cohesion of *integrated circuit*, *electric current* and *capacitor filters* is calculated as (30), (31) and (32) respectively:

(30)

$$C_T(\text{integr circuit}) = \frac{1}{1} \times 1 = 1$$

(31)

$$C_T(\text{electr current}) = \frac{3}{3} \times\ 1.58496 = 1.58496$$

(32)

$$C_T(\text{capacit filter}) = \frac{1}{1.73205} \times 1 = 0.57735$$

We can definitely see that this metric provides a more adequate ranking of the ngrams than C-value. Finally, cohesion values are normalized in a manner similar to those of relevance:

(33)

$$C_T(g) = 1 - \frac{1}{\log_2\left(2 + \frac{f_T(g)}{\sqrt[|g|]{\prod_{k_i \in g} f_T(k_i)}} \times F\right)}, \text{iff } |g| > 1$$

3.4. SRC: A Composite Measure for Term Extraction

Some studies (e.g. Zhang et al. 2008; Fedorenko et al. 2013) demonstrated that the combination of multiple term recognition algorithms tends to outperform most of the methods that consider only one statistical feature. On the other hand, a standard practice in the combination of measures (c.f. Frantzi et al. 2000; Park et al. 2002; Sclano and Velardi 2007; Lossio-Ventura et al. 2014) is to multiply each one of the weights derived from each algorithm by a different coefficient in order to provide a greater or lesser weight to the outcome of a given measure. Thus, the weighted composite measure for term extraction in DEXTER, which I call SRC, is as follows:

(34)

$$SRC_T(g) = termhood(g) + unithood(g)$$

$$termhood(g) = S_T(g) * \alpha + R_T(g) * \beta$$

$$unithood(g) = \begin{cases} 0, & \text{iff } |g| = 1 \\ C_T(g) * \gamma, & \text{iff } |g| > 1 \end{cases}$$

The fact that coefficients α, β and γ are user-adjustable constant values, providing that α + β = 1 for unigrams and α + β + γ = 1 for complex ngrams, is motivated by the fact that "domains and quality of corpus do have an impact on the performance of ATR algorithms" (Zhang et al. 2008, 2111). In this way, and depending on the values assigned to these coefficients when extracting complex ngrams, you have the choice of giving a greater weight to unithood (i.e. cohesion) rather than to termhood (i.e. salience and relevance), thus minimizing the problem that "the significance of unithood measures are often overshadowed by the larger notion of termhood" (Wonget al. 2008, 503).

4. Evaluation Procedure

Section 3 has revealed that the integration of the metrics related to the notions of salience, relevance and cohesion is theoretically well-grounded. Now it is time to evaluate empirically whether the composite measure SRC contributes to the performance of the ATE system. In this regard, it should be recalled that the goal of this article focuses on the suitability of the metric. In other words, if too much noise is found at the top of the candidate list, then the user will spend too much time removing false candidates, and this waste of time will definitely impact on the efficiency of the system.

As stated in Section 2, term extraction tools usually consider both linguistic and statistical properties of words for the identification of term candidates. Therefore, the precision of the ATE system is not only affected by the metric itself but also by other factors such as the granularity of linguistic patterns or the length of the stop-word list. In the case of DEXTER, full POS tagging is not performed, which seems quite logical considering that traditional linguistic patterns are quite restrictive as well as being language dependent. Instead, a set of shallow lexical filters supported by a stop-word

list is applied before term weighting. On the one hand, the system can filter out those candidates which match any of the following filters:

(i) Ngrams containing one or more non-alphabetical characters
(ii) One-character unigrams
(iii) Unigrams matching a functional word
(iv) Bigrams and trigrams containing a one-character word
(v) Bigrams containing one or two functional words
(vi) Trigrams containing a functional word at the beginning and/or at end of the ngram
(vii) Trigrams containing a non-prepositional functional word in the mid-position

On the other hand, DEXTER is provided with two stop-word lists for each language: (a) a pre-defined list of functional words, and (b) an automatically-generated list of other non-content bearing words which are derived on the basis of Luhn's word frequency distribution in a general corpus of the given language (Luhn 1958; Sun et al. 1999). Therefore, it can be assumed that a comparative evaluation of DEXTER with other term extraction tools can only serve to assess the performance of the whole system but not the real impact of the metric on this process. For these reasons, verifying that the combined measure performs better than each of its components taken separately in the same work environment is sufficient to empirically evaluate the suitability of SRC.

The experiments on SRC were performed with corpora of different sizes, domains and languages, and a hybrid procedure of evaluation was applied. The first stage of the evaluation was based on a reference list. First, those candidates extracted from the specialized corpora which were also present in their corresponding domains within the multilingual term database IATE (InterActive Terminology for Europe) were automatically tagged as positive candidates.[14] Second, positive candidates were reviewed to detect those which had been ill-categorized by IATE contributors. Finally, the second stage of the evaluation consisted in validating manually those candidates

[14] The IATE database was downloaded on 25 June 2014 from http://iate.europa.eu/tbxPageDownload.do.

which were not selected as terms in the previous stage. The experiments were aimed at comparing the performance of SRC, S, R and C in terms of precision from the top 200 unigrams, bigrams and trigrams.

4.1. Experiment 1

Experiment 1 was performed with a medium-sized corpus of 499 English-written documents (520,383 tokens) about electronics, whose corresponding IATE domains were Electrical Industry [6621001] and Electronics and Electrical Engineering [6826, 6826001, 6826002].[15] The documents were obtained from a website[16] whose aim is to provide beginners who study electronics with basic information to help them develop knowledge and understanding of this subject. DEXTER extracted 2,412 unigrams, 2,968 bigrams and 892 trigrams. According to the equation (34), the overall precision of SRC was 0.72.

Table 1 shows the results of the evaluation of unigrams, where SRC was calculated with $\alpha = 0.7$ and $\beta = 0.3$.

Table 1. Precision in the evaluation of unigrams

#candidates	**SRC**	**S**	**R**
1-50	0.90	0.84	0.74
1-100	0.82	0.82	0.71
1-150	0.80	0.78	0.72
1-200	0.78	0.77	0.72

In the case of bigrams, the best precision of SRC was achieved with the values 0.4, 0.2 and 0.4 for the coefficients α, β and γ respectively, as shown in Table 2.

Table 2. Precision in the evaluation of bigrams

[15] IATE domain codes are given in brackets.
[16] http://www.allaboutcircuits.com

#candidates	SRC	S	R	C
1-50	0.76	0.74	0.64	0.80
1-100	0.68	0.73	0.64	0.69
1-150	0.70	0.70	0.62	0.68
1-200	0.71	0.70	0.59	0.67

Finally, Table 3 illustrates the results of the evaluation of trigrams, where SRC had the values 0.6, 0.1 and 0.3 for the coefficients α, β and γ respectively.

Table 3. Precision in the evaluation of trigrams

#candidates	SRC	S	R	C
1-50	0.74	0.68	0.46	0.74
1-100	0.70	0.68	0.59	0.68
1-150	0.66	0.65	0.59	0.64
1-200	0.66	0.64	0.58	0.63

4.2. Experiment 2

Experiment 2 was performed with a small corpus of 137 Spanish texts (197,812 tokens) about medical procedures and surgery, whose corresponding IATE domains were Health [2841], Health Care Profession [2841001], Health Policy [2841002], Illness [2841003], and Medical Science [2841004]. The documents were obtained from *Enciclopedia Ilustrada de Salud ADAM*.[17] DEXTER extracted 1,707 unigrams, 504 bigrams and 278 trigrams. According to the equation (34), the overall precision of SRC was 0.89.

Table 4 illustrates the results of the evaluation of unigrams, where SRC was calculated with $\alpha = 0.9$ and $\beta = 0.1$.

Table 4. Precision in the evaluation of unigrams

[17] http://www.nlm.nih.gov/medlineplus/spanish/encyclopedia.html

#candidates	SRC	S	R
1-50	0.94	0.90	0.88
1-100	0.95	0.89	0.90
1-150	0.94	0.88	0.92
1-200	0.94	0.87	0.86

Table 5 shows the results of the evaluation of bigrams, where SRC was calculated with $\alpha = 0.4$, $\beta = 0.4$ and $\gamma = 0.2$.

Table 5. Precision in the evaluation of bigrams

#candidates	SRC	S	R	C
1-50	0.96	0.80	0.92	0.94
1-100	0.93	0.78	0.89	0.89
1-150	0.93	0.76	0.86	0.88
1-200	0.93	0.74	0.85	0.88

Finally, Table 6 illustrates the results of the evaluation of trigrams, where SRC has the values 0.8, 0 and 0.2 for the coefficients α, β and γ.

Table 6. Precision in the evaluation of trigrams

#candidates	SRC	S	R	C
1-50	0.84	0.78	0.82	0.82
1-100	0.84	0.79	0.81	0.78
1-150	0.80	0.80	0.80	0.78
1-200	0.79	0.77	0.74	0.76

4.3. Discussion of Results

Experiments 1 and 2 demonstrated that the best precision was obtained with SRC when the top 200 unigrams, bigrams and trigrams were retrieved. Indeed, if we analyze the above tables by rows, we can realize that on just two occasions single metrics improved the precision of SRC in the 24 different cut-off points along the top 200 ngrams, i.e. with the first 50 and 100 bigrams in Experiment 1. Therefore, it can be concluded that the combination of metrics can actually enhance the performance of the ATE system.

The key feature for the success of SRC is undoubtedly centered on the values assigned to the three coefficients in the equation (34). Indeed, SRC can become the best or the worst metric depending on these values. For example, if I had been interested in extracting just the top 50 bigrams in Experiment 1, the most efficient metric would have been C, unless I had tuned the SRC formula by adjusting the three coefficients. In fact, many researchers agree that:

> (…) it is reasonable to expect that there will be no "best" ATR [Automatic Term Recognition] method which would outperform others on all data sets and in all circumstances. (Knoth et al. 2009, 84)

However, on the basis of the theoretical underpinnings described in Section 3 and the experiments presented in this section, I can assert that, for ATE systems focusing on small- and medium-sized specialized corpora, SRC is one of the most efficient measures as far as precision is concerned. However, it is also reasonable to expect that there will be no pre-defined combination of constant values which would outperform others on all data sets and in all circumstances.

Consequently, how can one determine automatically the coefficient values which are able to outcome the largest number of terms among the top-ranked candidates in a given corpus? This is actually a challenge that I intend to face in future

experiments. At first sight, an attempt to manage this task would be based on the discriminating capacity of the S, R and C metrics compared to the distribution of the terms. This approach does not succeed, however, when looking at the distribution of bigrams in Experiment 2 (cf. Table 5); here the worst single metric was S and the best single metric was C, so it could be reasonably expected that their corresponding coefficients will be assigned a lower and higher value respectively in the SRC formula, but paradoxically $\alpha = 0.4$ and $\gamma = 0.2$. Consequently, there is not always a direct correlation between the weights of the single metrics and the single coefficients of the combined metric.

On the other hand, a more promising avenue of research is to discover the weights of the SRC parameters by automatically validating candidates with IATE and combining the different values of the single metrics to get the best distribution of terms in the different cut-off points along the top 200 ngrams. In this context, a given combination of the coefficients should be understood as an ordered selection of *k* elements from a set of *n* elements. It is important to note that the order of the selected elements matters, since an arrangement such as 0.8-0.2 is unlikely to provide the same outcome as 0.2-0.8 for α and β respectively. Therefore, mathematically speaking, these arrangements cannot be called combinations *in stricto sensu* but variations with repetition. However, these variations possess a special characteristic: the addition of the selected elements must be 1. Thus, being *n* the maximum number of values that each parameter can take (i.e. n = 11, since the range of values for any SRC parameter is from 0.0 to 1.0) and *k* the number of parameters (i.e. k = 2 for unigrams and k = 3 for bigrams and trigrams), the equation to calculate the number of permutations is as follows:

(35)

$$V_{n,k} = n, \text{iff } k = 2$$

$$V_{n,k} = \frac{n(n+1)}{2}, \text{iff } k = 3$$

Therefore, the ATE system requires 11 variations of the coefficients to calculate the best distribution of unigrams; and in the case of bigrams or trigrams, the variations are 66. However, this method is not fully reliable, since IATE is not a fully-developed terminological database. The pending question is whether the best variations of coefficients will still provide the best distribution of terms after the manual validation of candidates.

5. Conclusion

In this article, I presented the suitability of the metric used in DEXTER, an online multilingual ATE workbench. This metric, which I called SRC, is based on three fundamental notions: (i) Salience, i.e. which indicates the uniqueness or prevalence of a term in the data collection, (ii) Relevance, i.e. which measures the tendency of term usage between a domain-specific corpus and a general-purpose one, and (iii) Cohesion, i.e. which quantifies the degree of stability of multi-word terms. SRC can be described as a hybrid method, not only because it combines the linguistic approach with the statistical one, but also because it combines an AKE measure (i.e. salience) with ATE measures (i.e. relevance and cohesion). Moreover, SRC enables varying degrees of unification between termhood and unithood: whereas the salience and relevance components of SRC serve to measure termhood, because terms are properties of both documents and domains, the unithood of complex terms is determined by cohesion.

Unlike for most ATE research, where metrics are indiscriminately combined to produce the best results, the integration of DEXTER's terminological features (i.e. salience, relevance and cohesion) was grounded on a rational basis before the evaluation of the metric. The evaluation of the experiments in Section 4 proved that SRC can outperform the results obtained by single metrics, and more particularly by those upon which SRC has been devised, i.e. TF-IDF (Salton and Buckley 1988), weirdness (Ahmad et al. 2000) and cohesion (Park et al. 2002). The experiments also demonstrated that the distribution of term candidates in the different cut-off points along the top 200 ngrams becomes more adequate with SRC, suggesting the possibility of automatically fixing a minimum threshold below which domain-specific terms would be barely present. However, this issue will be addressed in future research. As stated by Conrado et al. (2014), one of the main challenges of ATE systems is precisely to determine a threshold in the candidate term ranking.

Acknowledgements

Financial support for this research has been provided by the DGI, Spanish Ministry of Education and Science, grants FFI2011-29798-C02-01 and FFI2014-53788-C3-1-P.

References

Ahmad, Khurshid, Lee Gillam, and Lena Tostevin. 2000. "Weirdness Indexing for Logical Document Extrapolation and Retrieval (WILDER)." In *Proceedings of the 8th Text Retrieval Conference (TREC-8)*, ed. by E. M. Voorhees, and D. K. Harman, 717-724. Washington: National Institute of Standards and Technology.

Barrón-Cedeño, Alberto, Gerardo Sierra, Patrick Drouin, and Sophia Ananiadou. 2009. "An Improved Automatic Term Recognition Method for Spanish." In *Computational Linguistics and Intelligent Text Processing*, ed. by Alexander Gelbukh, 125-136. Berlin-Heidelberg: Springer.

Barthes, Roland. 1964. *Elements of Semiology*. New York: Hill and Wang.

Church, Kenneth Ward, and Patrick Hanks. 1990. "Word Association Norms, Mutual Information and Lexicography." *Computational Linguistics* 6 (1): 22–29.

Church, Kenneth Ward, William Gale, Patrick Hanks, and Donald Hindle. 1991. "Using Statistics in Lexical Analysis." In *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon,* ed. by Uri Zernik, 115-164. Hillsdale, NJ: Lawrence Erlbaum.

Collins WordBanks Online (2013) (http://www.collins.co.uk/page/Wordbanks+Online). Accessed 14 August 2015.

Conrado, Merley da Silva, Ariani Felippo, Thiago Salgueiro Pardo, and Solange Rezende. 2014. "A Survey of Automatic Term Extraction for Brazilian Portuguese." *Journal of the Brazilian Computer Society* 20 (12): 1-28. (http://www.journal-bcs.com/content/20/1/12). Accessed 14 August 2015.

Conrado, Merley da Silva, Thiago Salgueiro Pardo, and Solange Rezende. 2014. "The Main Challenge of Semi-Automatic Term Extraction Methods." In *Proceedings of the 11th International Workshop on Natural Language Processing and Cognitive Science*, 1-10, Venice.

Dunning, Ted. 1994. "Accurate Methods for the Statistics of Surprise and Coincidence." *Computational Linguistics* 19 (1): 61-74.

Fedorenko, Denis, Nikita Astrakhantsev, and Denis Turdakov. 2013. "Automatic Recognition of Domain-Specific Terms: an Experimental Evaluation." In *Proceedings of the 9th Spring Researcher's Colloquium on Database and Information Systems*, 15-23, Kazan.

Frantzi, Katerina, and Sophia Ananiadou. 1996. "Extracting Nested Collocations." In *Proceedings of the 16th International Conference on Computational Linguistics*, 41-46. Morristown: Association for Computational Linguistics.

Frantzi, Katerina, Sophia Ananiadou, and Mima Hideki. 2000. "Automatic Recognition of Multi-Word Terms: the C-Value/NC-Value Method." *International Journal of Digital Libraries* 3 (2): 115-130.

Golik, Wiktoria, Robert Bossy, Zorana Ratkovic, and Claire Nédellec. 2013. "Improving Term Extraction with Linguistic Analysis in the Biomedical Domain." *Research in Computing Science* 70: 157-172.

Graf, Rudolf F. 1999. *Modern Dictionary of Electronics*, 7th edition. Boston: Newnes.

Grefenstette, Gregory. 1994. *Explorations in Automatic Thesaurus Discovery*. Boston: Kluwer Academic.

Harris, Zellig. 1954. "Distributional Structure." *Word* 10 (23): 146-162.

Kageura, Kyo, and Bin Umino. 1996. "Methods of Automatic Term Recognition: A Review". *Terminology* 3 (2): 259-289.

Knoth, Petr, Marek Schmidt, Pavel Smrz, and Zdenek Zdráhal. 2009. "Towards a Framework for Comparing Automatic Term Recognition Methods." In *Proceedings of the 8th Annual Conference Znalosti*, 83-94. Bratislava: Informatics and Information Technology STU.

Korkontzelos, Ioannis, Ioannis Klapaftis, and Suresh Manandhar. 2008. “Reviewing and Evaluating Automatic Term Recognition Techniques.” In *Proceedings of the 6th International Conference on Advances in Natural Language Processing*, ed. by Bengt Nordström, and Aarne Ranta, 248-259. Berlin-Heidelberg: Springer.

Kraaij, Wessel, and Renée Pohlmann. 1996. “Viewing Stemming as Recall Enhancement.” In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 40-48, Zurich.

Lossio-Ventura, Juan Antonio, Clement Jonquet, Mathieu Roche, and Maguelonne Teisseire. 2014. “Biomedical Terminology Extraction: A New Combination of Statistical and Web Mining Approaches.” In *Proceedings of Journées Internationales d’Analyse Statistique des Données Textuelles*, 1-12, Paris.

Luhn, Hans Peter. 1958. “The Automatic Creation of Literature Abstracts”. *IBM Journal of Research and Development* 2 (2): 159-165.

Mairal-Usón, Ricardo, and Carlos Periñán-Pascual. 2009. “The Anatomy of the Lexicon within the Framework of an NLP Knowledge Base.” *Revista Española de Lingüística Aplicada* 22: 217-244.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. 2009. *Introduction to Information Retrieval*. Cambridge: Cambridge University Press.

Nagao, Makoto, Mikio Mizutani, and Hiroyuki Ikeda. 1976. “An Automated Method of the Extraction of Important Words from Japanese Scientific Documents.” *Transactions of Information Processing Society of Japan* 17 (2): 110-117.

Navigli, Roberto, and Paola Velardi. 2002. “Semantic Interpretation of Terminological Strings.” In *Proceedings of the 6th International Conference on Terminology and Knowledge Engineering*, 95-100. Berlin-Heidelberg: Springer.

Park, Youngja, Roy J. Byrd, and Branimir K. Boguraev. 2002. “Automatic Glossary Extraction: Beyond Terminology Identification.” In *Proceedings of the 19th International Conference on Computational Linguistics*, vol. 1, 1-7. Stroudsburg, PA: Association for Computational Linguistics.

Pazienza, Maria Teresa, Marco Pennacchiotti, and Fabio Massimo Zanzotto. 2005. “Terminology Extraction: An Analysis of Linguistic and Statistical Approaches”. In *Studies in Fuzziness and Soft Computing: Knowledge Mining*, ed. by Janusz Kacprzyk, and Spiros Sirmakessis, 255-279. Berlin-Heidelberg: Springer.

Peñas, Anselmo, Felisa Verdejo, and Julio Gonzalo. 2001. “Corpus-Based Terminology Extraction Applied to Information Access.” In *Proceedings of the Corpus Linguistics Conference*, 458-465, Lancaster.

Periñán Pascual, Carlos. 2013. “A Knowledge-Engineering Approach to the Cognitive Categorization of Lexical Meaning.” *VIAL: Vigo International Journal of Applied Linguistics* 10: 85-104.

Periñán-Pascual, Carlos, and Francisco Arcas-Túnez. 2004. “Meaning Postulates in a Lexico-Conceptual Knowledge Base.” In *Proceedings of the 15th International Workshop on Databases and Expert Systems Applications*, 38-42. Los Alamitos: the Institute of Electrical and Electronics Engineers-Computer Society.

Periñán-Pascual, Carlos, and Francisco Arcas-Túnez. 2007. “Cognitive Modules of an NLP Knowledge Base for Language Understanding.” *Procesamiento del Lenguaje Natural* 39: 197-204.

Periñán-Pascual, Carlos, and Francisco Arcas-Túnez. 2010. “The Architecture of FunGramKB.” In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, 2667-2674. Malta: ELRA.

Periñán-Pascual, Carlos, and Ricardo Mairal-Usón. 2009. “Bringing Role and Reference Grammar to Natural Language Understanding.” *Procesamiento del Lenguaje Natural* 43: 265-273.

Plante, Pierre, and Lucie Dumas. 1989. “Le Dépouillement Terminologique Assisté par Ordinateur.” *Terminogramme* 46: 24-28.

Real Academia Española. *Corpus de Referencia del Español Actual (CREA).* (http://www.rae.es). Accessed 14 August 2015.

Sabbah, Yousef W., and Yousef Abuzir. 2005. “Automatic Term Extraction Using Statistical Techniques: A Comparative in-Depth Study & Applications.” In *Proceedings of the International Arab Conference on Information Technology ACIT 2005*, 1-7, Amman.

Sager, Juan C. 1990. *A Practical Course in Terminology Processing*. Amsterdam: John Benjamins.

Salton, Gerard, ed. 1971. *The SMART Retrieval System – Experiments in Automatic Document Retrieval*. Englewood Cliffs, NJ: Prentice Hall.

Salton, Gerard, and Christopher Buckley. 1988. “Term-Weighting Approaches in Automatic Text Retrieval.” *Information Processing & Management* 24 (5): 513-523.

Salton, Gerard, Anita Wong, and Chung-Shu Yang. 1975. “A Vector Space Model for Automatic Indexing.” *Communications of the ACM* 18 (11): 613-620.

Salton, Gerard, and Chung-Shu Yang. 1973. “On the Specification of Term in Automatic Indexing.” *Journal of Documentation* 29 (4): 351-372.

Salton, Gerard, Chung-Shu Yang, and Clement T. Yu. 1975. “A Theory of Term Importance in Automatic Text Analysis.” *Journal of the American Society for Information Science* 26 (1): 33-44.

Sclano, Francesco, and Paola Velardi. 2007. “TermExtractor: A Web Application to Learn the Common Terminology of Interest Groups and Research Communities.” In *Proceedings of the 9th Conference on Terminology and Artificial Intelligence*, 1-10, Sophia Antinopolis.

Singhal, Amit. 1997. *Term Weighting Revisited*. Ph.D. thesis. Ithaca, NY: Cornell University.

Singhal, Amit, Chris Buckley, and Mandar Mitra. 1996. “Pivoted Document Length Normalization.” In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 21-29. New York: ACM press

Singhal, Amit, Gerard Salton, and Chris Buckley. 1996. “Length Normalization in Degraded Text Collections.” In *Proceedings of the 5th Annual Symposium on Document Analysis and Information Retrieval*, 149-162. Las Vegas: University of Nevada.

Smadja, Frank. 1993. “Retrieving Collocations from Text: Xtract.” *Computational Linguistics* 19 (1): 143-178.

Sun, Qinglan, Debora Shaw, and Charles H. Davis. 1999. “A Model for Estimating the Occurrence of Same-Frequency Words and the Boundary between High- and Low-Frequency Words in Texts”. *Journal of the American Society for Information Science* 50 (3): 280-286.

The British National Corpus (BNC). Oxford University Computing Services. http://www.natcorp.ox.ac.uk

Turney, Peter D., and Patrick Pantel. 2010. “From Frequency to Meaning: Vector Space Models of Semantics.” *Journal of Artificial Intelligence Research* 37: 141-188.

Velardi, Paola, Michele Missikoff, and Roberto Basili. 2001. “Identification of Relevant Terms to Support the Construction of Domain Ontologies.” In *Proceedings of the Workshop on Human Language Technology and Knowledge Management*, 1-8. Morristown: Association for Computational Linguistics.

Wong, Wilson, Wei Liu, and Mohammed Bennamoun. 2007. “Determining Termhood for Learning Domain Ontologies Using Domain Prevalence and Tendency.” In *Proceedings of the 6th Australasian Conference on Data Mining*, 47-54, Gold Coast.

Wong, Wilson, Wei Liu, and Mohammed Bennamoun. 2008. “Determination of Unithood and Termhood for Term Recognition.” In *Handbook of Research on Text and Web Mining Technologies*, ed. by Min Song, and Yi-Fang Wu, 500-529. Hershey-New York: IGI Global.

Zhang, Ziqi, José Iria, Christopher Brewster, and Fabio Ciravegna. 2008. “A Comparative Evaluation of Term Recognition Algorithms.” In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, 2108-2113. Marrakech: ELRA.

Author's address

Carlos Periñán-Pascual
Applied Linguistics Department
Universitat Politècnica de València
Paranimf, 1 - 46730 Gandia
Valencia, Spain
jopepas3@upv.es

About the author

Carlos Periñán-Pascual studied English Language and Literature at Universitat de València and received his Ph.D. degree in English Philology at UNED in Madrid. Since his doctoral dissertation on the resolution of word-sense disambiguation in machine translation, his main research interests have included knowledge engineering, natural language understanding and computational linguistics. As a result, he has been the director and founder of the FunGramKB project since 2004, whose main goal is to develop a lexico-conceptual knowledge base to be implemented in NLP systems requiring language comprehension. After the design of the knowledge base, he also developed some tools for the FunGramKB Suite, such as the terminology extractor and the inference engine. His scientific production includes about 50 peer-reviewed publications in the fields of linguistics, natural language processing and artificial intelligence. He is currently an associate professor in the Applied Linguistics Department at Universitat Politècnica de València, Spain.