# DEXTER: a workbench for automatic term extraction with specialized corpora

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

## Abstract

Automatic term extraction has become a priority area of research within corpus processing. Despite the extensive literature in this field, there are still some outstanding issues that should be dealt with during the construction of term extractors, particularly those oriented to support research in terminology and terminography. In this regard, this article describes the design and development of DEXTER, an online workbench for the extraction of simple and complex terms from domain-specific corpora in English, French, Italian and Spanish. In this framework, three issues contribute to placing the most important terms in the foreground. First, unlike the elaborate morphosyntactic patterns proposed by most previous research, shallow lexical filters have been constructed to discard term candidates. Second, a large number of common stopwords are automatically detected by means of a method that relies on the IATE database together with the frequency distribution of the domain-specific corpus and a general corpus. Third, the term-ranking metric, which is grounded on the notions of salience, relevance and cohesion, is guided by the IATE database to display an adequate distribution of terms.

**Keywords**: terminology, terminography, automatic term extraction, DEXTER

## 1. *Introduction*

Nowadays, specialized-knowledge acquisition cannot be conceived without the use of a corpus. Discovering the lexical units of a given domain is undoubtedly a complex task where sheer introspection, or even the simple analysis of concordances, is often not an effective method.[1] Indeed, standard frequency criteria only serve to extract a general-purpose vocabulary, thus contributing little to the identification of technical words. This article falls within the field of automatic term extraction (ATE), which represents a priority area of interest for most industry-of-language service providers. Indeed, ATE has two major applications, i.e. as a tool for research and professional purposes (e.g. terminologists, translators, interpreters, etc.) or as a component of a knowledge-based system (e.g. document classification, information retrieval, text summarization, etc.). The first application is the focus of this investigation, and more particularly in relation to corpus-based terminology and terminography research. Therefore, our interest lies in the development of software for (a) the study of words and phrases that pertain to particular areas of specialized knowledge, and (b) the compilation of domain-specific lexical units to create language resources such as glossaries or term databases. From this approach, an in-depth study of the state of the art in ATE revealed a number of

[1] For the sake of clarity, we restrict the use of "word" to the narrower notion of orthographic word, so an ngram is a sequence of *n* words. Moreover, "lexical unit" refers to a unit of meaning that is realized by one or more words, resulting in single-word (simple) and multi-word (complex) lexical units. Whereas all lexical units are composed of ngrams (e.g. *porphyria cutanea tarda* consists of three unigrams, two bigrams and one trigram), not all ngrams can be considered as lexical units (e.g. *cutanea tarda*). Finally, lexical units can appear in general or specialized domains. In this latter case, we use "term" to refer to a lexical unit that is characteristically associated with a given scientific or technical domain. For example, *porphyria cutanea tarda* is a medical term.

overarching requirements that newly-developed term extractors should meet. It should be noted that, although it is not unusual for existing ATE systems to meet some of the following requirements, all of the requirements can rarely be found in the same system:

Requirement 1. The system should make use of an adequate statistical measure. Ideally, “statistical adequacy” would involve that the system could extract all and only the true terms of the domain. Unfortunately, this assumption is so ambitious that it is unlikely to be realistic. On the one hand, although reference lists as gold standards are available, it is not reliable to evaluate the performance of ATE systems in terms of recall, since "such resources are far from exhaustive and not error free" (Vivaldi and Rodríguez 2007: 237). Consequently, if recall is to be evaluated, then all term candidates have to be checked manually, which becomes a time-consuming task. On the other hand, the effectiveness of probabilistic measures is closely dependent on the intrinsic characteristics of the corpus. Therefore, the adequacy of the metric should be reasonably understood as the possibility that the system manages the peculiarities of different corpora by outperforming other metrics. In this regard, one of the main purposes of term extractors is to significantly reduce the amount of noise in the ranked list, since "the more time users spend in scanning through term candidate lists, the less useful the tool is” (Thurmair 2003: 6). Thus, a typical way to measure the statistical adequacy consists in verifying that little noise (i.e. few false candidates) has been generated with the top-ranked terms output by the term extractor.

Requirement 2. The system should extract simple and complex terms. Although most of the works in this area still rely on multi-word terminological units (cf. Deane 2005; Wermter and Hahn 2005; among many others), “one cannot deny the fact that simple terms have a role to play” (Wong, Liu and Bennamoun 2008: 503). In fact, Zhang, Iria, Brewster and Ciravegna (2008) showed that single-word terms can be as important as multi-word units and occupy a fairly large proportion in certain domains, so “algorithms that ignore single-word terms may cause problems to tasks built on top of ATR [Automatic Term Recognition]” (Zhang *et al*. 2008: 2108).

Requirement 3. The system should extract the nouns, verbs and adjectives that are used to describe a given specialized domain. Most ATE systems have focused exclusively on noun phrases, under the assumption that they make up the bulk of the terminological inventory. As explained by Justeson and Katz (1995: 9), "judging from data in dictionaries of technical vocabulary, the majority of technical terms do consist of more than one word; among these, the overwhelming majority are noun phrases, which constitute the vast majority of multi-word terminological units in probably all domains". Although this assumption still has a deep effect on some current research experiments (cf. Paulo and Mamede 2004; Pazienza, Pennacchiotti and Zanzotto 2005; Ittoo, Maruster, Wortmann and Bouma 2010; Merkel, Foo and Ahrenberg 2013; Zadeh and Handschuh 2014a; Meyers, He, Glass and Babko-Malaya 2015; among others), it is also true that “verbs and adjectives, though they have received much less attention, can also be domain-specific” (Ahrenberg 2009).

Requirement 4. The system should be able to be extended to other languages and other scientific/technical domains. It is reasonable to assume that this adaptability can be favoured if statistical knowledge-poor methods of term extraction are employed, which is in line with the natural language processing (NLP) techniques used in information retrieval, where simple methods such as stopwords and stemming usually yield more significant improvements than higher-level processing, e.g. chunking or parsing, which increase processing and even decrease precision (Brants 2004). However, this approach breaks away from that of mainstream term extractors, where a hybrid approach combines statistical methods with linguistic methods, where part-of-speech (POS) tagging plays a crucial role.

Requirement 5. The system should be provided with a user-friendly interface that can fully integrate the built-in functionalities of corpus management, term extraction, and term management, which are required for terminology and terminography research. In other words, if you want to go further than a local experiment with a toy implementation of the term extractor, you need a number of tools that can support the complete workflow of term processing in a single platform.

In this context, this research led to the design and development of DEXTER (Discovering and EXtracting TERminology), an online workbench for the recognition, validation and management of the simple and complex terms (namely, unigrams, bigrams and trigrams) extracted from non-structured texts in small- and medium-sized specialized corpora in English, French, Italian and Spanish.[2] DEXTER not only integrates all of the requirements outlined above but also introduces some improvements in the main modules of the ATE process (i.e. preprocessing, extraction, recognition, weighting and validation), since "although ATE has been researched for more than 20 years, there is still room for improvement" (Conrado, Felippo, Pardo and Rezende 2014: 1). The remainder of this article is structured as follows. Section 2 describes the different stages involved in ATE with DEXTER. Section 3 deals with the evaluation of DEXTER in comparison with BioTex, GaleXtract, Termine and TermoStat. Finally, Section 4 concludes with a review of the five requirements from DEXTER's perspective.

## 2. *Automatic term extraction in DEXTER*

[2] DEXTER, which has been developed in C# with ASP.NET 4.0, is freely accessible from the FunGramKB website (http://www.fungramkb.com/nlp.aspx).

### 2.1 *Graphical user interface application*

DEXTER has been developed as a Web-based user-interface application, which is not confined to term extraction in the strict sense but provides an environment suitable for linguistic research, with functionalities such as corpus compilation and textual exploration, among many others. Here the ATE process consists of a pipeline of five stages: (1) corpus registration, (2) corpus development, (3) candidate extraction, including term recognition, (4) term weighting, and (5) term validation and clean-up. To illustrate the explanations in this and other sections, an experiment was conducted with an English corpus of 200 documents (312,710 tokens), where 1,452 unigrams, 2,143 bigrams and 711 trigrams were extracted as term candidates. This collection of documents, which we will call "sample corpus", was downloaded from a website whose aim is to provide students with basic information about electronics.[3]

#### 2.1.1 *Corpus registration*

The domain-specific corpus is manually tagged with descriptors such as the name (e.g. Electronics corpus), the language (e.g. English), the content description (e.g. Tutorials for engineering students on all aspects of basic electronics), and the true and false domain(s), which are selected from the IATE database (InterActive Terminology for Europe). With respect to the latter, users must select the "true domains", i.e. the most relevant field(s) of specialized knowledge described in the corpus (e.g. Electrical Industry [6621001] and Electronics and Electrical Engineering [6826, 6826001, 6826002]),[4] and can also select some "false domains", which serve to discard term candidates that, although they are likely to occur in the corpus, pertain to other specialized domains (e.g. Chemistry [6811, 6811001, 6811002, 6811004], Computer

[3] http://www.electronics-tutorials.ws

[4] IATE domain codes are given in brackets.

Science [3236001, 3236002], Mechanical Engineering [6821, 6821001, 6821002, 6821003] and Science [36]). True and false domains play an important role in both term recognition and term weighting.

### 2.1.2 *Corpus development*

This stage consists of two actions. First, the corpus is compiled. DEXTER has been designed for small and medium-sized corpora (i.e. up to one million tokens). As stated by Koester (2010: 68-69), specialized corpora do not need to be "as large as more general corpora to yield reliable results"; since specialized corpora are "carefully targeted, they are more likely to reliably represent a particular register or genre than general corpora". Another feature of the corpus is its sample size, which is closely related to the issue of representativeness. For example, "a corpus of a million words or so cannot afford to include whole books which might be up to 100,000 words in length" (Hunston, 2008: 165), since such large documents could result in a disproportionate composition of the corpus. Although the suitability of a sample size depends on the specific task that is undertaken, "experience with samples of 20,000 words has shown that on the whole these are sufficiently large to yield statistically reliable results on frequency and distribution" (Haan 1992: 3), which helped us determine the maximum number of tokens in a single document (i.e. 25,000 tokens). Apart from plain-text files, PDF documents and HTML Web pages can also be uploaded. Each document can also be manually tagged with descriptors such as the title (e.g. Transistor biasing) and the content description (e.g. Transistor base biasing configurations from a single supply available for a common emitter amplifier).

Second, corpus indexation is performed with Lucene.Net (Hatcher, Gospodnetic and McCandless 2010)—the C# port of Lucene, one of the most popular open-source library for high-performance information retrieval. Document indexation enables the user to retrieve the context of all the words derived from a given stemmed ngram. A naïve approach of searching a certain word would have been to sequentially scan each text file for the target word, but this rudimentary method could have generated a bottleneck in case of a large collection of documents.

### 2.1.3 *Candidate extraction and term recognition*

This stage consists of four actions. First, each document in the corpus is preprocessed. Each document is tokenized by a simple analyzer, which splits tokens at non-letter characters and then lowercases each token, discarding numbers from the token stream. Then, the tokens are processed by the Snowball stemmer, and unigrams, bigrams and trigrams are derived from the stems. The default frequency threshold is set to three. As usual, a threshold is applied not only to reduce noise but also "to avoid producing a too long list that might become a hindrance for the experts evaluating the output" (Marín 2015: 7).

Second, term recognition is performed with IATE, i.e. the identification of known terms by comparing the list of candidates with a term database. Today there are a number of terminology data banks that are both multidisciplinary and multilingual, such as *Le grand dictionnaire terminologique,*[5] *TERMIUM*[6] or *TermSciences*.[7] For a platform such as DEXTER, one of the best options of gold standard is IATE, because it provides a large coverage of words (i.e. 8.5 million terms) from a high number of domains and subdomains (i.e. 21 domains that are hierarchically structured into 649 subdomains) in a high number of languages (i.e. 24 languages). The main goal of this term database,

[5] http://www.granddictionnaire.com
[6] http://www.btb.termiumplus.gc.ca/tpv2alpha/alpha-eng.html
[7] http://www.termsciences.fr

which can be downloaded in the XML-based *TermBase eXchange* (TBX) format,[8] is to ensure the quality of the terminology employed in the documentation of the EU institutions. In this stage, DEXTER can recognize (a) a large number of extracted ngrams that pertain to the true domains of the corpus, which are automatically tagged as positive candidates, and (b) those ngrams that pertain to the false domains, which are eliminated from the term-candidate list. False domains enable DEXTER to deal effectively with multi-domain discourse, which is particularly common in scientific and technical writing. For instance, in the sample corpus, the system recognized many lexical units typically used in other specialized domains, such as *cadmium sulphide* (chemistry), *star network* (computer science), *root mean square* (mathematics) or *absolute position encoder* (mechanical engineering). As a result, to make the list of ngrams more domain-focused (i.e. electronics), the stems pertaining to these false domains (i.e. chemistry, computer science, mathematics and mechanical engineering) were discarded just before term weighting, providing that the stem did not belong to some true domain.

Third, DEXTER makes use of a predefined list of functional stopwords (e.g. *for, in, the*, etc.). Moreover, the system dynamically generates a list of common stopwords (e.g. *allow, high, know, type*, etc.) that is tailor-made for the domain-specific corpus.[9] Whereas stopword lists are usually obtained from general corpora (cf. Fox 1990) or elaborated for a few scientific domains in a given language (cf. Jacquey, Tutin, Kister, Jacques, Hatier and Ollinger 2013), DEXTER is provided with a multilingual and multi-domain method to automatically discover common stopwords in specialized corpora, as explained in Section 2.3.1.

[8] The IATE database was downloaded from http://iate.europa.eu/tbxPageDownload.do.

[9] For the purpose of brevity, we use the phrase "common stopwords" to refer to the non-functional topic-neutral words found in a given collection of documents.

Finally, a set of lexical filters, which are applied to the stemmed ngrams that were not recognized as relevant domain-specific terms in the second step of this stage, serves to discard many candidates for further analysis. For example, the following types of stemmed ngrams are ignored:

a) Ngrams containing one single character (e.g. *D* in “In the above circuit, node D is chosen as…”)

b) Ngrams containing one or more non-alphabetical characters (e.g. *V1* in “Consider two AC voltages, V1 having a peak voltage of …”)

c) Unigrams matching a functional or common stopword (e.g. *call* in “These audio signal transformers are called ‘matching transformers’…”)

d) Unigrams derived from words starting with a capital letter, provided that they do not belong to the general domain (e.g. *Schmitt* in “…the higher threshold value of the Schmitt trigger…”); this unsophisticated procedure is intended to identify domain-specific named entities, so a more powerful named-entity recognizer will be used in future versions.

e) Bigrams containing one or two functional stopwords (e.g. *the AC* in “…an important parameter of the AC waveform…”)

f) Bigrams or trigrams containing at least one common stopword (e.g. *short length* in “…a short length of wire designed to…”)

g) Trigrams containing a functional stopword at the beginning and/or the end of the ngram (e.g. *the form of* in “…in the form of infra-red radiation.”)

h) Trigrams containing a functional stopword in the mid-position, provided that the stopword is not a preposition (e.g. *volt or amp* in “…the signal waveform measured in volts or amps.”)

It should be noted that these shallow lexical filters cannot be equated to standard grammatical patterns used for term-candidate detection (cf. Justeson and Katz 1995), since the former do not actually require that the content of the corpus be annotated by POS taggers or parsers. As a result, the above lexical filters can be extended to other languages more easily than grammatical-pattern matching techniques, which are heavily language-dependent. Therefore, DEXTER does not adopt a traditional hybrid approach to term extraction, since the linguistic properties of words are not taken into account.

### 2.1.4 *Term weighting*

DEXTER employs a parameterized composite metric called SRC (Periñán-Pascual 2015), which is grounded on the notions of salience (S), relevance (R) and cohesion (C):

(1)

$$SRC(g) = S(g) * \alpha + R(g) * \beta + C(g) * \gamma$$

where *g* is a stemmed ngram, and α, β and γ are user-adjustable coefficients. The user can specify the value of each of the three coefficients, or the system itself can discover the “best” weights for these parameters. The SRC metric and its application for term extraction are described in more detail in Section 2.3.2.

### 2.1.5 *Term validation and candidate clean-up*

Term weighting outputs an inventory of SRC-ranked candidates that still requires that users eliminate irrelevant ngrams. As stated by Oakes (1998), deleting false positives is a much simpler task than creating the whole list of terms manually, and this validation task becomes much easier when some of the terms have already been recognized with the help of a term database (e.g. IATE). For this reason, it can be said that DEXTER adopts a hybrid approach to the evaluation of term candidates. The semi-automatic process of term evaluation is further described in Section 2.3.3.

### 2.2. *Web service*

DEXTER has also been developed as a SOAP/WSDL-based Web service through which you can upload a zipped corpus of TXT documents, together with information about the language and the true and false domains of the corpus. In this case, the system provides a limited set of the features available in the graphical user interface (GUI) application, resulting in an ATE method that consists of only two stages: (1) candidate extraction, including term recognition, and (2) term weighting, as described in Figure 1. Both stages were briefly outlined above. Detailed descriptions of the actions [E] and [G] are given in Section 2.3.1 and Section 2.3.2 respectively.

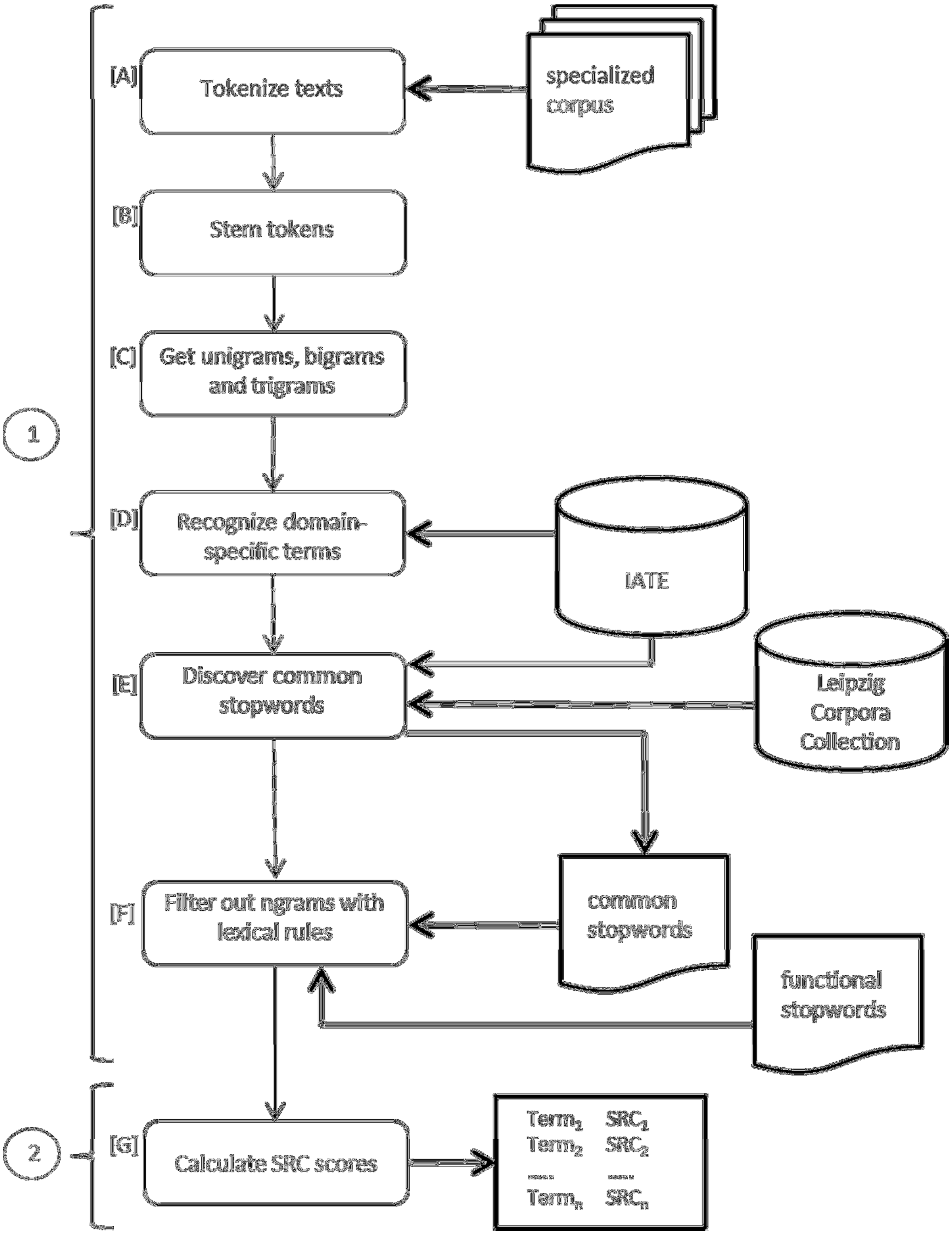


Figure 1. DEXTER Web service: workflow of term extraction.

2.3. *Some issues on DEXTER*

After a brief description of the stages involved in the term-extraction process within DEXTER, this section provides a detailed account of its most innovative aspects.

2.3.1. *The removal of stopwords*

A stopword list is a valuable resource in text-data analysis in general, and in information retrieval, information extraction, document clustering and document categorization in particular. Stopwords are “words having no significant semantic relation to the context in which they exist” (Khosrow-Pour 2009: 3112), so they should be removed before processing starts in order to achieve greater effectiveness. The selection of this type of words actually becomes a crucial factor in term extraction, since “proper choice and construction of a stoplist can affect results in a way that depends on the task at hand” (Sinka and Corne 2003: 401). However, despite their importance, stopwords are not so carefully selected as would be expected.

In this search for non-significant words, stopword lists usually consist of two categories of words: (a) functional words, and (b) common words. The task to discover functional words is a non-issue, since they can be easily obtained from the grammar of the language. For example, in the case of (a) in DEXTER, 454 stems were detected for English, including not only functional words (e.g. articles, conjunctions, determiners, prepositions and pronouns, among others) but also Arabic and Roman numerals and common abbreviations used in professional and academic documents (e.g. *i.e., c.f., etc* or *et al*). In the case of (b), however, there is no generally-accepted list of non-informative words for every language. There are a number of stopword lists scattered over the Web, but they tend to include primarily functional words. Moreover, stopword lists have been traditionally hand-crafted from the author's experience. This situation is further compounded by data obsolescence, as illustrated by one of the most popular stopword lists (cf. Fox 1990), which was defined from the Brown Corpus of English (Francis and Kučera 1982) more than twenty-five years ago. In this context, we developed an adaptive method to automatically identify stopwords in domain-specific corpora. This method is described in the remainder of this section.

In the last decade, most of the NLP research including some ATE component has employed stopword lists that were constructed from one of the following methods:

- a multilingual and multi-domain method, which automatically generates stopword lists for different languages by means of entropy-based metrics applied to specialized corpora (cf. Sinka and Corne 2003; Zou, Wang, Deng, Han and Wang 2006; Alajmi, Saad and Darwish 2012; Asubiaro 2013),
- a monolingual and multi-domain method, which provides a stopword list derived from the general corpus of a given language; in this regard, Salton (1971) and Fox (1990) have been the most popular lists for English (cf. Karystianis, Buchan and Nenadic 2014; Zadeh and Handschuh 2014b; Sajjacholapunt and Joy 2015; Conde, Larrañaga, Arruarte, Elorriaga and Roth 2016),[10] or
- a monolingual and single-domain method, where the stopword list is intended for a given specialized domain in a given language (cf. Jacquey, Tutin, Kister, Jacques, Hatier and Ollinger 2013), thus substantially restricting the scope of applicability.

Our interest lies primarily in the first type of method, since this research has been developed in a multilingual environment that was designed to process document collections from a wide variety of scientific and technical domains. As shown below, however, a more effective approach to stopword detection is required.

Research on the automatic generation of stopword lists usually aims to measure the importance of a term within a document collection by means of entropy. In this

[10] A repository of general stopword lists for various languages can be found at http://members.unine.ch/jacques.savoy/clef/index.html, where the lists for other languages than English were constructed following the guidelines described in Fox (1990).

regard, the method typically used to calculate the normalized value of entropy is that of Lochbaum and Streeter (1989: 672):

(2)

$$entropy_k = 1 - \frac{noise_k}{\log_2 NDocs}$$

where noise is calculated in turn following Salton and McGill (1983: 65):

(3)

$$noise_k = \sum_{i=1}^{NDocs} \frac{Freq_{ik}}{Totalfreq_k} \log_2 \frac{Totalfreq_k}{Freq_{ik}}$$

where *NDocs* equals the total number of documents, $Freq_{i,k}$ is the frequency of the *k*th term in the *i*th document, and $Totalfreq_k$ is the total frequency with which $term_k$ occurs in all documents of the corpus. Therefore, the higher entropy the word has, the less information the word is likely to reveal, and so it is more likely to be seen as a stopword. In the equation (3), noise measures the concentration of a given term rather than occurrence counts (Harman 1986), i.e. words with high frequency in many documents of the collection will have low entropy. However, following the tf-idf metric (Salton and Buckley 1988), which measures the importance of a word in a collection of documents, when a given term occurs in many documents, it has a low discriminating power. Therefore, entropy calculations are biased towards unduly low entropies for general-purpose words that are frequent in a large number of documents within a domain-specific corpus. For example, this problem can be found with the words that have the lowest entropy values in the sample corpus, including technical terms such as *circuit* (0.07819), *voltage* (0.09429) and *current* (0.11117), but also common words such as *use* (0.04712), *look* (0.08761), *call* (0.09287) and *give* (0.10010).

As can be concluded from the above, entropy is not a reliable metric to discriminate common stopwords in specialized corpora, so we chose to develop a method based on the distribution of word frequencies in a general corpus. In particular, we used Sun, Shaw and Davis's (1999: 285) transition point as "a threshold value establishing a mark between high-frequency and low-frequency words", which can be predicted with the following equation:

(4)

$$n = \sqrt{D}$$

where $D$ is the number of distinct word forms in a document collection. As "the frequency of word occurrence in an article furnishes a useful measurement of word significance" (Luhn 1958: 160), this transition point can be used to distinguish between significant and non-significant words, thus contributing to the recognition of common stopwords in domain-specific corpora. To this end, a word-frequency list derived from a general corpus was provided for each of the languages in the system. In this case, the Leipzig Corpora Collection (Quasthoff, Richter and Biemann 2006; Biemann, Heyer, Quasthoff and Richter 2007) became a highly valuable resource, since it contains corpora of similar size and content in 110 languages. More specifically, we downloaded the 1-million-sentence corpora of English, French, Italian and Spanish that were compiled in 2010 from newspaper texts.[11] For the sake of clarity, the stopword identification process is illustrated only with the English corpus, which contains 16,949,229 tokens and 114,229 word forms. However, the same tasks were performed with the other general corpora. Furthermore, to facilitate understanding of the method, the DEXTER database scheme can be formally characterized as follows:

(5)

[11] http://corpora.uni-leipzig.de/download.html

$$\text{DEXTER}_{ske} := \left\{ \begin{array}{l} \left(\text{CORPUS}_{gen};\{\text{STEM,FREQ}\}\right), \\ \left(\text{CORPUS}_{spe};\{\text{STEM,FREQ}\}\right), \\ \left(\text{STOP}_{fun};\{\text{STEM}\}\right), \\ \left(\text{IATE}_{con};\{\text{ID\_CONCEPT,ID\_DOMAIN}\}\right), \\ \left(\text{IATE}_{ter};\{\text{ID\_CONCEPT,STEM}\}\right) \end{array} \right\}$$

A database scheme D := {R$_1$, …, R$_n$} is usually defined as a set of relation schemes, where a relation scheme R;{A$_1$, …, A$_n$} consists of a finite set of attributes. For example, CORPUS$_{\text{gen}}$ and CORPUS$_{\text{spe}}$ refer to the general corpus (46,128 stems) and the specialized corpus (3,330 stems) respectively, STOP$_{\text{fun}}$ is the list of functional stopwords (454 stems), and IATE$_{\text{con}}$ and IATE$_{\text{ter}}$ hold 1,842,937 concepts and 256,502 English stemmed unigrams categorized by the IATE database. Moreover, given S ⊆ R, let $t$↓{S} denote the restriction of a tuple $t$ over R on S. For example, if the relation scheme is CORPUS$_{\text{gen}}$;{STEM, FREQ} and $g$ = (transistor, 1196) is a tuple over CORPUS$_{\text{gen}}$, then $g$↓{STEM} = (transistor). The remaining symbols in (5)-(10) are used in the standard notation for set theory and symbolic logic. The complexity of the actual database design is certainly underspecified in the scheme (5), which includes only those relations that are relevant for the issues of this section.

The first point to make is that DEXTER takes the stem as the minimum unit of processing, so the inventory of word forms in the general corpus was reduced to 46,128 stems.[12] Thus, CORPUS$_{\text{gen}}$ held these stems and their frequencies. It was found that the transition point $n$ was 215 for this distribution of stemmed unigrams, so DEXTER selected all the non-functional stems whose frequency was within the range from $n$ back to rank 1. Formally, this can be expressed as in the function (6), i.e. the value of STEM in every tuple $g$ of CORPUS$_{\text{gen}}$ where FREQ in $g$ is greater or equal to 215 and there

[12] Named entities were removed from consideration.

exists no tuple *f* in $\mathrm{STOP}_{\mathrm{fun}}$ that contains the value of STEM, where $\mathrm{COMMON}_{\mathrm{gen}}$ returned 4,195 stems.

(6)

$$\mathrm{COMMON}_{gen} := \left\{ \begin{array}{l} g \downarrow \{\mathrm{STEM}\} \mid g \in \mathrm{DEXTER}\left(\mathrm{CORPUS}_{gen}\right) \wedge \\ g\left(\mathrm{FREQ}\right) \geq 215 \wedge \\ \neg\left(\exists f \in \mathrm{DEXTER}\left(\mathrm{STOP}_{fun}\right) : f\left(\mathrm{STEM}\right) = g\left(\mathrm{STEM}\right)\right) \end{array} \right\}$$

On the other hand, the same procedure was applied to the sample corpus (i.e. $\mathrm{CORPUS}_{\mathrm{spe}}$), where *n* was 64. In this case, $\mathrm{COMMON}_{\mathrm{spe}}$ returned 358 stems, as described in the function (7), i.e. the value of STEM in every tuple *s* of $\mathrm{CORPUS}_{\mathrm{spe}}$ where FREQ in *s* is greater or equal to 64 and there exists no tuple *f* in $\mathrm{STOP}_{\mathrm{fun}}$ that contains the value of STEM.

(7)

$$\mathrm{COMMON}_{spe} := \left\{ \begin{array}{l} s \downarrow \{\mathrm{STEM}\} \mid \mathrm{s} \in \mathrm{DEXTER}\left(\mathrm{CORPUS}_{spe}\right) \wedge \\ s\left(\mathrm{FREQ}\right) \geq 64 \wedge \\ \neg\left(\exists f \in \mathrm{DEXTER}\left(\mathrm{STOP}_{fun}\right) : f\left(\mathrm{STEM}\right) = s\left(\mathrm{STEM}\right)\right) \end{array} \right\}$$

It should be noted that this adaptive procedure to discover common stopwords does only take into account stemmed unigrams, because they are the ones used as the building blocks in lexical filters, which in turn allow the system to discard not only unigrams but also bigrams and trigrams before term weighting.

At this point, it is worthwhile to recall that, when the sample corpus was registered, some descriptors were provided (cf. Section 2.1.1). For example, the most-relevant IATE domains manually chosen for the sample corpus were Electrical Industry [6621001], Electronics and Electrical Engineering [6826], Electrical Engineering [6826001] and Electronics Industry [6826002]. Therefore, to complete the remaining tasks, it was necessary to identify the 6,863 stems pertaining to such "true domains" in

IATE, as described in the function (8), i.e. the value of STEM in every tuple *t* of IATE$_{\text{ter}}$ where there exists a tuple *c* in IATE$_{\text{con}}$ whose ID_CONCEPT is the same as ID_CONCEPT in *t* and ID_DOMAIN in *c* is 6621001, 6826, 6826001 or 6826002.

(8)

$$\text{IATE}_{electro} := \left\{ \begin{array}{l} t \downarrow \{\text{STEM}\} \mid \text{t} \in \text{DEXTER}(\text{IATE}_{ter}) \\ \wedge \left( \begin{array}{l} \exists c \in \text{DEXTER}(\text{IATE}_{con}): \\ c(\text{ID\_CONCEPT}) = t(\text{ID\_CONCEPT}) \wedge \\ c(\text{ID\_DOMAIN}) \in \left\{ \begin{array}{l} 6621001, 6826, \\ 6826001, 6826002 \end{array} \right\} \end{array} \right) \end{array} \right\}$$

At this stage, the system has five sets available—i.e. CORPUS$_{\text{gen}}$, COMMON$_{\text{gen}}$, CORPUS$_{\text{spe}}$, COMMON$_{\text{spe}}$, and IATE$_{\text{electro}}$, where $\text{COMMON}_{gen} \subset \text{CORPUS}_{gen}$ and $\text{COMMON}_{spe} \subset \text{CORPUS}_{spe}$. The core issue now is to decide how these sets can be used to recognize most of the common stopwords in the sample corpus (i.e. STOP$_{\text{com}}$). In this regard, two possible methods can be applied, which return the sets consisting of all the stems that occur in both COMMON$_{\text{gen}}$ and the set resulting from the stems of COMMON$_{\text{spe}}$ (9a) or CORPUS$_{\text{spe}}$ (9b) that do not belong to IATE$_{\text{electro}}$.

(9a)

$$\text{STOP'}_{com} = \text{COMMON}_{gen} \cap \left(\text{COMMON}_{spe} - \text{IATE}_{electro}\right)$$

(9b)

$$\text{STOP''}_{com} = \text{COMMON}_{gen} \cap \left(\text{CORPUS}_{spe} - \text{IATE}_{electro}\right)$$

More specifically, these two methods correspond to the functions (10a) and (10b) respectively, which return the value of STEM in every tuple *r* of COMMON$_{\text{spe}}$ (10a) or CORPUS$_{\text{spe}}$ (10b) providing that there exists a tuple *g* in COMMON$_{\text{gen}}$ that contains the value of STEM but there exists no tuple *i* in IATE$_{\text{electro}}$ that contains the value of STEM.

(10a)

$$\mathrm{STOP'}_{com} := \left\{ r \downarrow \{\mathrm{STEM}\} \left| \begin{array}{l} r \in \mathrm{DEXTER}\left(\mathrm{COMMON}_{spe}\right) \wedge \\ \left( \begin{array}{l} \exists g \in \mathrm{DEXTER}\left(\mathrm{COMMON}_{gen}\right): \\ g\left(\mathrm{STEM}\right) = r\left(\mathrm{STEM}\right) \end{array} \right) \wedge \\ \neg \left( \begin{array}{l} \exists i \in \mathrm{DEXTER}\left(\mathrm{IATE}_{electro}\right): \\ i\left(\mathrm{STEM}\right) = r\left(\mathrm{STEM}\right) \end{array} \right) \end{array} \right. \right\}$$

(10b)

$$\mathrm{STOP''}_{com} := \left\{ r \downarrow \{\mathrm{STEM}\} \left| \begin{array}{l} r \in \mathrm{DEXTER}\left(\mathrm{CORPUS}_{spe}\right) \wedge \\ \left( \begin{array}{l} \exists g \in \mathrm{DEXTER}\left(\mathrm{COMMON}_{gen}\right): \\ g\left(\mathrm{STEM}\right) = r\left(\mathrm{STEM}\right) \end{array} \right) \wedge \\ \neg \left( \begin{array}{l} \exists i \in \mathrm{DEXTER}\left(\mathrm{IATE}_{electro}\right): \\ i\left(\mathrm{STEM}\right) = r\left(\mathrm{STEM}\right) \end{array} \right) \end{array} \right. \right\}$$

Both methods are based on the same two premises. On the one hand, it can be assumed that the stopwords in the sample corpus should be derived from the top-ranked words in the general corpus. The next issue to be considered is the search space in the sample corpus, i.e. $\mathrm{COMMON}_{\mathrm{spe}}$ in the function (10a) or $\mathrm{CORPUS}_{\mathrm{spe}}$ in the function (10b). Indeed, this is a critical decision, since the effectiveness of term extraction is closely related to the aggressivity of candidate-space reduction by stopwords. For example, the functions (10a) and (10b) resulted in $|\mathrm{STOP'}_{\mathrm{com}}| = 114$ and $|\mathrm{STOP''}_{\mathrm{com}}| = 942$ respectively. Both sets were manually reviewed for evaluation, whose results are displayed in Table 1.

| common stopwords | precision | recall | F-score |
|---|---|---|---|
| $\mathrm{STOP'}_{\mathrm{com}}$ | 0.98245 | 0.12055 | 0.21212 |
| $\mathrm{STOP''}_{\mathrm{com}}$ | 0.98619 | 1.00000 | 0.99305 |

Table 1. Evaluation of stopwords for DEXTER.

It should be noticed that recall was evaluated by taking into account all stems in both stopword lists that were not related to the given specialized domain; considering that $\text{STOP}'_{com} \subseteq \text{STOP}''_{com}$, then the latter has no false negatives. Table 1 demonstrates that STOP''$_{\text{com}}$, i.e. the most aggressive stopword-removal method, was actually the most effective one.

On the other hand, it is also reasonable to believe that stopwords can only be found among the stems that do not belong to any of the IATE true domains of the sample corpus. This assumption raises a non-trivial issue, since neonyms are not the only type of term to appear in domain-specific corpora. According to ISO 704 (2009), terms are formed by applying one of the following methods: (a) creating neoterms (i.e. newly coined technical or scientific words and phrases), (b) using forms that exist in the general language (e.g. through processes such as conversion, terminologization, semantic transfer and transdisciplinary borrowing), or (c) translingual borrowing (i.e. existing terms in one language can be introduced into another language). Therefore, it was found that terminologized words such as *face, state* or *turn* (Table 2), which were all present in both COMMON$_{\text{gen}}$ and CORPUS$_{\text{spe}}$, became a major source of error that was eliminated by the term database. Indeed, if the IATE-based component of the methods (9a) and (9b) had been ignored, then |STOP'$_{\text{com}}$| = 263 and |STOP''$_{\text{com}}$| = 1438, but the error rates would have been 0.57414 and 0.35396 respectively.

| term | Definition from IATE (Electronics and Electrical Engineering) |
|---|---|
| face | the transparent end of the cone through which the image is viewed or projected |
| state | the assigned range of voltage, current, etc., corresponding to one of the distinct recognisable conditions of a digital signal |
| turn | a basic coil element which forms a single conducting loop |

comprising one insulated conductor

Table 2. Instances of terminologization.

It should be recalled that the goal is not to automatically build a stopword list for a given specialized domain but only to discover the stopwords that are present in a given domain-specific corpus. To this end, the next task was to evaluate this adaptive method against the static method of applying a list of 571 stopwords used in the SMART information retrieval system (Salton 1971) and a list of 421 stopwords proposed by Fox (1990). In this case, the number of stems from non-functional words was 942 for DEXTER$_{stop\_com}$ (or STOP''$_{com}$), 163 for SMART$_{stop\_com}$ and 89 for FOX$_{stop\_com}$. Assuming that the system should ideally detect all and only those stems in these three lists that are not relevant to the sample corpus, the evaluation of DEXTER, SMART and FOX provided the results shown in Table 3.

| common stopwords | precision | recall | F-score |
|---|---|---|---|
| DEXTER | 0.98619 | 0.90634 | 0.94458 |
| SMART | 0.98773 | 0.15707 | 0.23869 |
| FOX | 0.96629 | 0.08390 | 0.14333 |

Table 3. Evaluation of stopword lists.

This experiment demonstrates that our stopword-removal method is not only the most aggressive but also the most effective, i.e. high precision and recall with respect to other stopword lists.

Finally, we can assess the impact that DEXTER$_{stop\_com}$ has on the processing of the sample corpus. In this regard, we highlight the capacity of our method to substantially reduce the number of irrelevant term candidates, contributing ultimately to the increase in precision. It should be recalled that two of the lexical filters used by DEXTER are based on common stopwords; particularly, (a) the unigrams that match a common stopword and (b) the bigrams or trigrams that contain at least one common stopword are eliminated. Suppose that neither of these two lexical filters is applied. In this case, the inventory of candidates would have increased by 39.34% for unigrams (i.e. from 1,452 to 2,394 candidates), 27.03% for bigrams (i.e. from 2,143 to 2,937 candidates) and 50.03% for trigrams (i.e. from 711 to 1,423 candidates). Figure 2 shows the distribution of the common stopwords as unigrams or as part of bigrams and trigrams in the sample corpus. In the range [1-200], for example, 24 unigrams (e.g. *calculate, equation* or *symbol*) would have been recognized as common stopwords, and 64 bigrams (e.g. *common use, gate symbol* or *previous tutorial*) and 97 trigrams (e.g. *amount of current, ideal for use, nearest preferred value* or *temporary storage device*) would have consisted of one or more common stopwords. Considering that none of these unigrams and bigrams and only eight of the trigrams (i.e. *current-limiting resistor, free-wheeling diode, high-pass filter, high-wave rectifier, liquid-crystal display, low-pass filter, magneto-motive force* and *push-pull amplifier*) are true terms, these stopwords would have contributed 0.12, 0.32 and 0.44 to the false-discovery rate in the top-ranked 200 unigrams, bigrams and trigrams respectively, affecting the precision in this range of candidates.

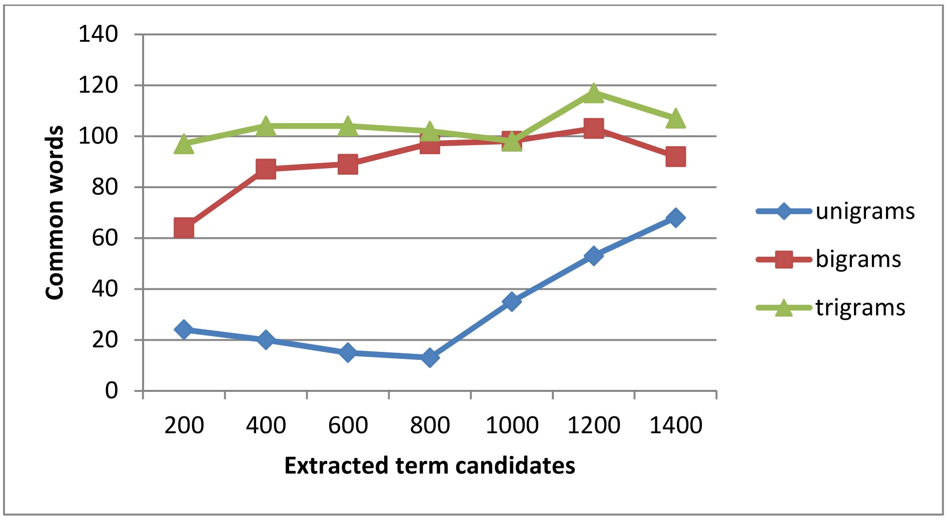


Figure 2. Distribution of common words.

Therefore, it can be concluded that DEXTER is provided with an effective stopword-detection method that can be adapted not to a few large areas of knowledge (e.g. economics, law, science or technology) but to the whole range of highly-specialized fields of expertise (e.g. civil law, accounting system, space science or mechanical engineering) in twenty-four languages that are represented in IATE.

### 2.3.2. *The SRC metric*

The research literature has demonstrated that the combination of multiple ATE algorithms tends to outperform most of the methods that consider only one statistical feature (cf. Zhang *et al*. 2008; Fedorenko, Astrakhantsev and Turdakov 2013). Indeed, the recent trend does not seek to devise new measures for unithood and/or termhood but to combine statistical features effectively, where:

> '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. (Kageura and Umino 1996: 260-261)

Moreover, a metric with adjustable parameters allows the system to accommodate to the configuration of the document collection. However, unlike for most of the research in ATE, where single metrics are usually combined indiscriminately to produce the best results, SRC is grounded on the theoretical principles of salience, relevance and cohesion. Periñán-Pascual (2015) presented a detailed description of these terminological features.

On the one hand, one of the pillars of SRC is the notion of salience, which is based on the termhood measure tf-idf (Salton, Wong and Yang 1975; Salton, Yang and Yu 1975; Salton and Buckley 1988), 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 whole document collection (or inverse document frequency, i.e. idf). This decision is supported by the fact that the task of automatic document indexation has a clear point in common with that of automatic term extraction, since the keywords employed to index a given domain-specific document are usually perceived as terminological units (Pazienza *et al.* 2005). In this regard, the two most popular weighting measures in automatic keyword extraction are tf-idf and Okapi BM25 (Robertson, Walker and Beaulieu 1998). Thus, the salience of the stemmed ngram *g* in the document *d* is calculated in DEXTER by applying the following formula:

(11a)

$$S_d(g) = TF(g) * IDF(g) * NORM(g)$$

(11b)

$$TF(g) = f_d(g)$$

(11c)

$$IDF(g) = 1 + \log\left(\frac{N_T}{df(g)}\right), \text{where } df(g) > 0$$

(11d)

$$NORM(g) = \frac{1}{\sqrt{\sum_{g \in d}\left(TF(g) \times IDF(g)\right)^2}}$$

where $f_d(g)$ is the number of occurrences of *g* in *d*, $N_T$ is the number of documents in the target corpus, and *df(g)* is the number of documents in which the ngram appears in the target corpus. The normalization factor, which makes the salience index range from 0 to 1, is calculated on the basis of the type of ngram. 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.

The rationale for the three components of the equation (11) 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 that 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 (a) long documents usually use the same ngrams repeatedly, and (b) long documents have numerous different ngrams (Singhal, Buckley and Mitra 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.

The salience of ngrams with respect to the whole target corpus, and not just to a single document, can be calculated as follows:

(12)

$$S(g) = \frac{\sum_{d \in CP_T} S_d(g)}{\sqrt{\sum_{g_j \in CP_T} \left(S(g_j)\right)^2}}$$

Again, the normalization factor of this formula only takes into account ngrams of the same type.

On the other hand, salience is combined with a measure that quantifies the relevance of ngrams through the contrastive analysis between the target corpus and a reference corpus. Salience measures the prevalence of the term in a particular target domain, but it does not serve to reflect the tendency of term usage across different domains, that is, the relevance of the term. In this context, relevance is calculated in DEXTER as follows, which results from an adaptation of Ahmad, Gillam and Tostevin's weirdness (2000):

(13a)

$$R(g)^{''} = \frac{P_T(g)}{P_R(g)}$$

(13b)

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

(13c)

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

where $f_T(g)$ and $f_R(g)$ represent the frequency of the stemmed ngram $g$ in the target corpus and the reference corpus respectively, $f_T(k)$ and $f_R(k)$ represent the frequency of a given lexical item in $g$ with respect to the target corpus and the reference corpus respectively, $|CP_T|$ and $|CP_R|$ represent the total number of words in the target corpus and the reference corpus respectively, and $|g|$ is the number of lexical items included in the ngram. In this setting, if an ngram is used more frequently in the domain of the target corpus than in the domain of the reference corpus, 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 $f_R(g) = 1$. The relevance of complex ngrams is calculated on the basis of the geometric mean of each lexical item within the ngram. In this way, the metric can minimize the effects of extremely small or large values in a skewed frequency distribution of the items within the multi-word ngram. The relevance index is normalized with the following equation:

(14)

$$R(g) = 1 - \frac{1}{\log_2\left(2 + R(g)''\right)}$$

Finally, the notion of cohesion was introduced to determine the unithood of complex ngrams. Therefore, whereas salience and relevance serve to measure termhood, cohesion is aimed at quantifying the degree of stability of bigrams and trigrams. Although many association measures have been employed for unithood, such as $\Phi^2$ (Gale and Church 1991), $\chi^2$ (Nagao, Mizutani and Ikeda 1976), cubic mutual information (Vivaldi, Màrquez and Rodríguez 2001), Dice coefficient (Smadja, McKeown and Hatzivassiloglou 1996), log likelihood (Dunning 1994), log odds ratio (Everitt 1992), mutual expectation (Silva, Dias, Guilloré and Lopes 1999), (pointwise) mutual information (Church and Hanks 1990), symmetric conditional probability (Silva and Lopes 1999) or t-score (Church, Gale, Hanks and Hindle 1991), Park, Byrd and Boguraev (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, Klapaftis and Manandhar (2008: 249) showed that:

> 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, Ananiadou and Mima 2000), which is calculated as follows:

(15a)

$$C\text{-value}(g) = \log_2|g| * f_T(g), \text{ iff } g \text{ is not nested}$$

(15b)

$$C\text{-value}(g) = \log_2|g| * (f_T(g) - NST(g)), \text{ otherwise}$$

(15c)

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

where |*g*| is the number of lexical items included in the stemmed ngram *g*, being |*g*| > 1, *$f_T(g)$* represents the frequency of *g* in the target corpus, and $S_g$ is the set of longer term candidates that contain *g*. On the one hand, |*g*| plays a key role: a longer ngram appearing *n* times in a corpus has a higher score 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. On the other hand, if *g* appears as nested, the equation (15c) serves to quantify nestedness as the degree of independence of *g*; in other words, the greater the number of longer ngrams in which *g* appears as nested, the smaller the independence of *g*.

However, C-value presents a problem in the weighting of bigrams. In particular, in case of a bigram not appearing as a nested sequence of longer ngrams, C-value is equal to the frequency of the bigram, since $\log_2(2) = 1$ and the NST component is not taken into consideration. In this context, C-value is incapable of adequately measuring the unithood of the bigrams that do not occur as part of longer ngrams in the target corpus, being unable to discriminate complex terms from recurrent co-occurrences of unigrams with their collocates. It should be noted that most of the research with C-value has focused on long term candidates in the medical domain, where 4-gram and even 5-gram candidates are relatively frequent, e.g. *adenoid cystic basal cell carcinoma* (cf. Frantzi *et al.* 2000). However, this degree of complexity is certainly infrequent in some other specialized domains.

In the light of this evidence, cohesion is calculated in DEXTER as follows, which results from an adaptation of Park *et al.*'s Term Cohesion (2002):

(16a)

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

(16b)

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

where $f_T(g)$ is the frequency of the stemmed ngram $g$ in the target corpus, $f_T(k)$ is the frequency of a given lexical item in $g$ with respect to the target corpus, and $|g|$ is the number of items in $g$. Cohesion takes into account both the frequency of $g$ and the frequency of the items that compose $g$. More particularly, cohesion is high when the items that compose the ngram are more frequently found within the ngram than alone in texts. In this line, we 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. As with the relevance metric, the geometric mean in (16a) smooths the result in a frequency distribution where extreme values can be present. Cohesion values are also normalized in a manner similar to those of relevance:

(17)

$$C(g) = 1 - \frac{1}{\log_2\left(2 + C(g)''\right)}, \text{iff } |g| > 1$$

Therefore, DEXTER integrates the above terminological features through the following SRC equation:

(18a)

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

(18b)

$$termhood(g) = S(g) * \alpha + R(g) * \beta$$

(18c)

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

where *g* is a stemmed ngram, and the coefficients α, β and γ are user-adjustable, providing that α + β = 1 for unigrams and α + β + γ = 1 for complex ngrams. As explained above, *S(g), R(g)* and *C(g)* outcome normalized values.

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, Schmidt, Smrz and Zdráhal 2009: 84), so it is also reasonable to expect that there will be no predefined combination of constant values in SRC which would outperform others on all data sets and in all circumstances.

In this regard, DEXTER can automatically discover the most suitable weights for the SRC coefficients after term recognition with IATE. Thus, taking into account all the permutations of these coefficients, the system gets the combination that provides (a) the highest precision with the top-ranked 200 ngrams and (b) the most gradual distribution of the terms with respect to four cut-off points (i.e. 50, 100, 150 and 200) along the top-ranked 200 ngrams. In particular, DEXTER examines eleven permutations of the coefficients to calculate the best distribution of unigrams; and in the case of bigrams or trigrams, the permutations are sixty-six. This task can be performed only when the ngrams found in the IATE true domains have been automatically tagged as positive candidates. For example, Table 4 shows how terms are distributed among the unigrams of the four ranges in the sample corpus before human validation.

| S-R | 1-50 | 51-100 | 101-150 | 151-200 | total |
|---|---|---|---|---|---|
| 1.0-0.0 | 49 | 45 | 45 | 40 | 179 |
| 0.9-0.1 | 48 | 41 | 40 | 31 | 160 |
| 0.8-0.2 | 46 | 37 | 34 | 28 | 145 |
| 0.7-0.3 | 46 | 33 | 24 | 27 | 130 |
| 0.6-0.4 | 39 | 30 | 26 | 22 | 117 |
| 0.5-0.5 | 37 | 26 | 22 | 22 | 107 |
| 0.4-0.6 | 37 | 22 | 20 | 20 | 99 |
| 0.3-0.7 | 35 | 21 | 20 | 19 | 95 |
| 0.2-0.8 | 32 | 20 | 19 | 19 | 90 |
| 0.1-0.9 | 29 | 20 | 20 | 19 | 88 |
| 0.0-1.0 | 26 | 21 | 21 | 14 | 82 |

Table 4. SRC coefficients and term distributions (sample corpus).

In this case, DEXTER easily discovers that the highest precision is obtained with the first combination in Table 4, i.e. [α = 1; β = 0]. However, when there is no single winning combination, the next step consists in determining which distribution shows the most gradual distribution of the terms along the top-ranked 200 ngrams.

The assumption of the gradual distribution of terms is based on what Pazienza *et al.* (2005: 270) described as "the power of each measure in discriminating true and false terms", i.e. true terms should be assigned to the highest positions in the rank, while the remaining false terms concentrate closer to the bottom of the list. Consequently, although the values of the SRC coefficients can be determined in part by the gradual distribution of only the top-ranked 200 candidates, a decreasing precision trend is expected to be shown in the remaining candidates, so that the discriminating power of the metric produces a significant ranking. In DEXTER, this notion of "gradual distribution of terms" plays a key role when the permutations of the values of the SRC coefficients do not result in a winning combination. One of the most logical choices was initially to use a measure based on the ranks of data. For example, non-parametric measures of rank correlation such as Gamma, Kendall and Spearman could have served to determine the strength of the relationship between the rank (i.e. the cut-off point) and the number of terms that were detected in each rank. However, we soon realized that these measures failed to achieve their goal. By way of example, suppose that DEXTER finds ten combinations of the α, β and γ values that yield the same highest precision, e.g. 140 terms have been recognized by IATE among the top 200 SRC-ranked ngrams. In this hypothetical scenario, Table 5 shows the scores derived from the rank correlation measures, where the resulting distributions have been labelled from A to J.

| | 1-50 | 51-100 | 101-150 | 151-200 | Gamma | Kendall | Spearman |
|---|---|---|---|---|---|---|---|
| A | 37 | 36 | 33 | 34 | -0.667 | -0.667 | -0.800 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| B | 36 | 37 | 34 | 33 | -0.667 | -0.667 | -0.800 |
| C | 34 | 34 | 34 | 38 | -1.000 | -0.707 | -0.775 |
| D | 32 | 36 | 36 | 36 | -1.000 | -0.707 | -0.775 |
| E | 37 | 34 | 33 | 36 | -0.333 | -0.333 | -0.400 |
| F | 37 | 33 | 36 | 34 | -0.333 | -0.333 | -0.400 |
| G | 36 | 34 | 37 | 33 | -0.333 | -0.333 | -0.400 |
| H | 34 | 37 | 36 | 33 | -0.333 | -0.333 | -0.400 |
| I | 36 | 32 | 36 | 36 | -0.333 | -0.236 | -0.258 |
| J | 34 | 34 | 38 | 34 | -0.333 | -0.236 | -0.258 |

Table 5. A hypothetical distribution of terms (rank correlation).

As can be noted in C, D, I and J, it is not uncommon to find two or more ranks with the same number of terms. Although extended measures were used to effectively manage tied ranks (e.g. Kendall's tau-b), none of the coefficients above showed sufficient discriminating power. For example, E, F, G and H have the same score with any of the three coefficients, so DEXTER is still not able to select a winning distribution of terms. The problem is that the calculation is not based on the actual values of the data points but only on the rank order.

As the gradual distribution of terms is based on the assumption that the bulk of terms is likely to be found at the top of the list and the number of specialized words progressively declines to the bottom, the alternative was to take into consideration the cumulative number of terms and precision in each cut-off point (i.e. 1-50, 1-100, 1-150 and 1-200), where a linear pattern of correlation is usually revealed. For example, Figure 3 displays the data points corresponding to the distributions A and B, whose best-fitting lines are represented by the dashed line and the dotted line respectively.

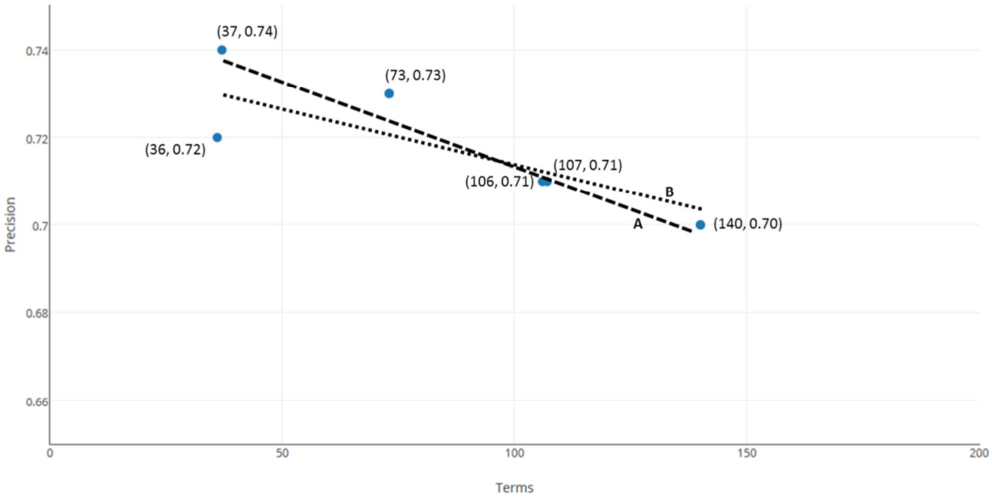


Figure 3. Two linear distributions of terms.

In this approach, the distribution of terms can be measured by means of Pearson’s product-moment correlation coefficient, since we are concerned with how closely the points fit to the line. Table 6 demonstrates that Pearson's correlation coefficient (*r*) can actually get better results with the distributions in Table 5.

| | 1-50 | 1-100 | 1-150 | 1-200 | r |
|---|---|---|---|---|---|
| A | 0.74 | 0.73 | 0.71 | 0.70 | -0.9881 |
| E | 0.74 | 0.71 | 0.69 | 0.70 | -0.8281 |
| B | 0.72 | 0.73 | 0.71 | 0.70 | -0.7859 |
| F | 0.74 | 0.70 | 0.71 | 0.70 | -0.7402 |
| G | 0.72 | 0.70 | 0.71 | 0.70 | -0.6625 |
| I | 0.72 | 0.68 | 0.69 | 0.70 | -0.3518 |
| H | 0.68 | 0.71 | 0.71 | 0.70 | 0.5671 |
| J | 0.68 | 0.68 | 0.71 | 0.70 | 0.7879 |
| C | 0.68 | 0.68 | 0.68 | 0.70 | 0.7919 |
| D | 0.64 | 0.68 | 0.69 | 0.70 | 0.9327 |

Table 6. A hypothetical distribution of terms (correlation coefficient).

Finally, a measure was devised whereby the adequacy of a given distribution of terms (*a*) is calculated primarily on precision (*p*) and to a lesser degree on the correlation coefficient (*r*), as shown in the equation (19), where *e* is the exponential constant.

(19)

$$a = e^{p} - \frac{1}{2}r$$

To conclude, Table 7 shows the scores assigned to the distributions of unigrams in Table 4.

| S-R | p | r | a |
|---|---|---|---|
| 1.0-0.0 | 0.90 | -0.9764 | 2.9478 |
| 0.9-0.1 | 0.80 | -0.9875 | 2.7193 |
| 0.8-0.2 | 0.72 | -0.9955 | 2.5522 |
| 0.7-0.3 | 0.65 | -0.9829 | 2.4070 |
| 0.6-0.4 | 0.58 | -0.9974 | 2.2847 |
| 0.5-0.5 | 0.54 | -0.9745 | 2.2033 |
| 0.4-0.6 | 0.50 | -0.9506 | 2.1240 |
| 0.3-0.7 | 0.47 | -0.9584 | 2.0792 |
| 0.2-0.8 | 0.45 | -0.9431 | 2.0399 |
| 0.1-0.9 | 0.44 | -0.9424 | 2.0239 |
| 0.0-1.0 | 0.41 | -0.9839 | 1.9988 |

Table 7. Scores of term distributions (sample corpus).

### 2.3.3. *Computer-aided term validation*

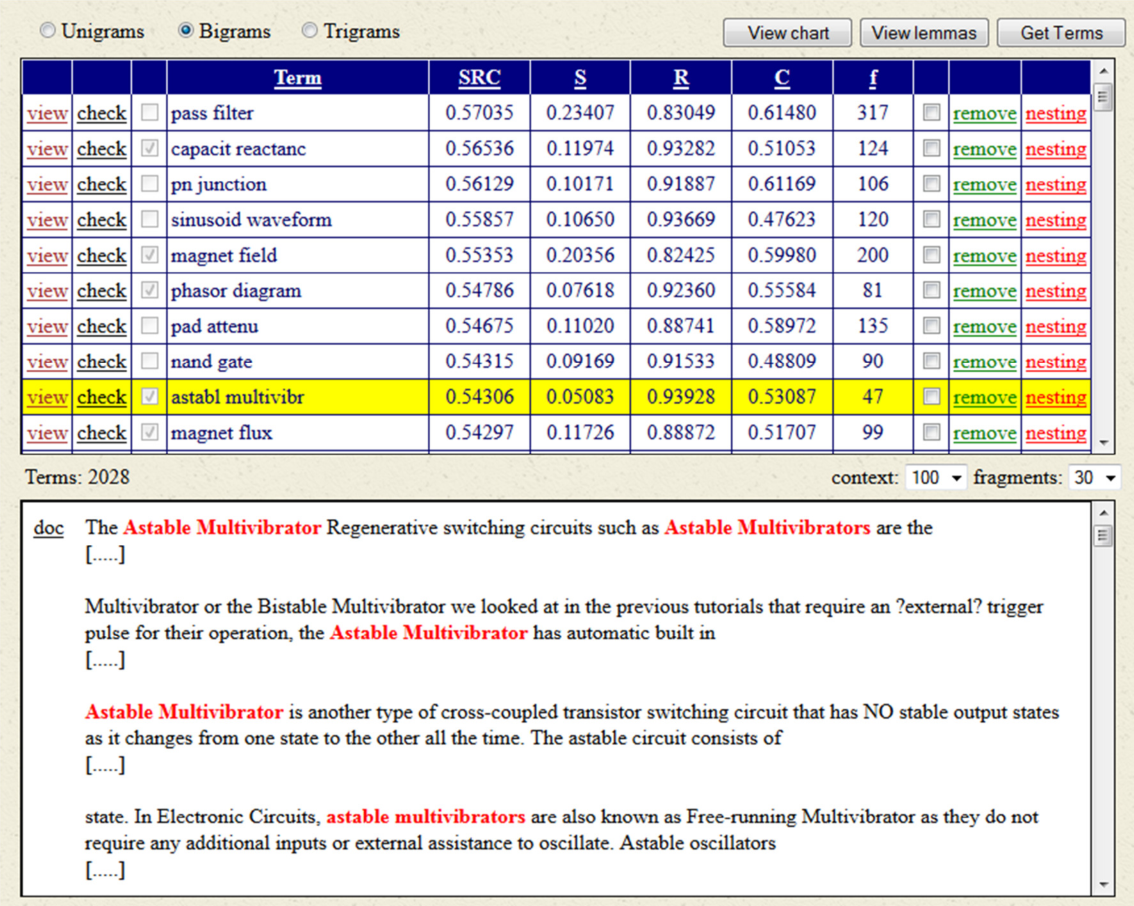
DEXTER adopts a hybrid approach to term evaluation, since term recognition based on the IATE database is integrated with human validation based on experts' judgements. As described by Pazienza *et al.* (2005: 265), the evaluation of ATE systems has been traditionally performed by means of one of two methods: (a) an *a priori* reference list of terms for the specific domain is used as a gold standard against which to measure the system performance, or (b) human experts validate the term candidates extracted by the system. It is important to highlight that both methods have drawbacks. With respect to (a), the system can extract terminological expressions which are not present in the reference list; in this case, although these candidates are true terms, they are treated as false negatives. With respect to (b), manual validation is a time-consuming task, as well as being prone to the expert's subjectivity and personal interpretation. DEXTER minimizes these problems by adopting both methods. The remainder of this section deals with the computer-assisted mode of data validation and clean-up.

Once the ngrams of a given type are ranked according to their SRC weight, their validation is performed from the most specific (i.e. trigrams) to the most generic (i.e. unigrams) in relation to their cognitive load. The reason is that the complexity of ngrams is proportional to their information content. For example, the information content of the unigram *championship* is lower than that of the bigram *football championship*, and in turn this bigram becomes less informative than the trigram *European football championship*. False candidates (i.e. true negatives and false positives) are manually discarded. A key issue at this stage of the validation is that DEXTER can provide the context of any term candidate, in the form of a maximum of eighty snippets for each candidate with a maximum of 400 characters per fragment. Moreover, the user can browse the whole document from where a given snippet has been extracted. Since DEXTER is based on the indexation and search capabilities of Lucene technology, context retrieval starts with an ngram-based query to the corpus, where every document that matches the query during the search is assigned a score computing how similar the document is to the query. Indeed, Lucene uses a formula that is primarily based on tf-idf, together with other factors such as coordination, field-length normalization, and term/query boosting, to calculate the relevance of matching documents;[13] then, DEXTER displays snippets of documents reverse-sorted by this score. It is noteworthy that, although most Key-Word-in-Context (KWIC) concordancing programs (cf. Wiechmann and Fuhs 2006) allow the user to determine the amount of co-text to the left and right of the keyword, they cannot guarantee the relevance of the data. It is certainly not a matter of size, i.e. the length of the concordance span, but rather of significance, i.e. if meaning arises within the concordance span. In this regard, the statistical significance of Lucene's similarity score helps to put in the foreground the most relevant contexts of ngrams.

Finally, candidate space can be reduced on the grounds of lack of term nestedness, that is, when one or more terms are embedded in a larger term. For example, in the case of bigrams and trigrams, you have the option of discarding all their nested ngrams automatically. It is important to make it clear that the degree of nestedness is variable, ranging from full to partial nestedness and to no nestedness at all. For example, *capacitor electrolyte* serves to exemplify full nestedness, since both *capacitor* and *electrolyte* are specialized terms. On the other hand, *alternating current* illustrates partial nestedness, since *alternating* is not an actual term in the electronics domain. Finally, *Thomson effect* does not have any nestedness, since neither *Thomson* nor *effect* is a domain-specific term. Therefore, as shown in Figure 3, DEXTER allows you to eliminate false candidates in two ways: (a) in case of partial or full nestedness, you only eliminate a given ngram by clicking "remove", but (b) in case of no nestedness, you can eliminate the false candidate together with all their embedded ngrams by clicking "nesting". Indeed, removing all embedded ngrams at one time certainly results in an

[13] Lucene's similarity scoring formula is described in Hatcher *et al.* (2010: 86-88).

effective clean-up method during manual validation, because the number of ngrams that can be discarded with just one click amounts to three with a bigram and to six with a trigram.



| | | | Term | SRC | S | R | C | f | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| view | check | □ | pass filter | 0.57035 | 0.23407 | 0.83049 | 0.61480 | 317 | □ | remove | nesting |
| view | check | ☑ | capacit reactanc | 0.56536 | 0.11974 | 0.93282 | 0.51053 | 124 | □ | remove | nesting |
| view | check | □ | pn junction | 0.56129 | 0.10171 | 0.91887 | 0.61169 | 106 | □ | remove | nesting |
| view | check | □ | sinusoid waveform | 0.55857 | 0.10650 | 0.93669 | 0.47623 | 120 | □ | remove | nesting |
| view | check | ☑ | magnet field | 0.55353 | 0.20356 | 0.82425 | 0.59980 | 200 | □ | remove | nesting |
| view | check | ☑ | phasor diagram | 0.54786 | 0.07618 | 0.92360 | 0.55584 | 81 | □ | remove | nesting |
| view | check | □ | pad attenu | 0.54675 | 0.11020 | 0.88741 | 0.58972 | 135 | □ | remove | nesting |
| view | check | □ | nand gate | 0.54315 | 0.09169 | 0.91533 | 0.48809 | 90 | □ | remove | nesting |
| view | check | ☑ | astabl multivibr | 0.54306 | 0.05083 | 0.93928 | 0.53087 | 47 | □ | remove | nesting |
| view | check | ☑ | magnet flux | 0.54297 | 0.11726 | 0.88872 | 0.51707 | 99 | □ | remove | nesting |

Terms: 2028

context: 100 fragments: 30

doc The **Astable Multivibrator** Regenerative switching circuits such as **Astable Multivibrators** are the
[.....]

Multivibrator or the Bistable Multivibrator we looked at in the previous tutorials that require an ?external? trigger pulse for their operation, the **Astable Multivibrator** has automatic built in
[.....]

**Astable Multivibrator** is another type of cross-coupled transistor switching circuit that has NO stable output states as it changes from one state to the other all the time. The astable circuit consists of
[.....]

state. In Electronic Circuits, **astable multivibrators** are also known as Free-running Multivibrator as they do not require any additional inputs or external assistance to oscillate. Astable oscillators
[.....]

Figure 4. Validation and clean-up in DEXTER.

It could be thought that a metric such as C-value, which is described as “a method to improve the extraction of nested terms” (Frantzi *et al.* 2000: 122), could contribute to automate part of the process of data validation. However, as explained in Section 2.3.2, C-value would not have been as efficient as expected with bigrams and trigrams in small- and medium-sized corpora.

## 3. *Evaluation of DEXTER*

The evaluation was conducted by comparing the results achieved by DEXTER with those obtained by BioTex (Lossio-Ventura, Jonquet, Roche and Teisseire 2014a),[14] GaleXtract (Barcala, Domínguez-Noya, Gamallo, López, Moscoso, Rojo, Santalla and Sotelo 2007),[15] Termine (Frantzi *et al.* 2000)[16] and TermoStat (Drouin 2003),[17] whose main features are described as follows:

- BioTex and TermoStat can discover simple and complex terms, whereas GaleXtract and Termine recognize just complex terms.
- Unlike Termine (English), BioTex (English, French, Spanish), GaleXtract (English, French, Galician, Portuguese, Spanish) and TermoStat (English, French, Italian, Portuguese and Spanish) are multilingual.
- BioTex, GaleXtract, Termine and TermoStat make use of TreeTagger for the POS-based filtering of term candidates.
- In BioTex, a system for biomedical term extraction, the user can change the number of linguistic patterns used to filter term candidates, as well as the function (i.e. average, maximum or sum) in the metrics F-Ocapi and F-TFIDF-C. In this research, BioTex was configured with the default number of linguistic patterns (i.e. 200) and with the maximum function, since Lossio-Ventura *et al.* (2014b) demonstrated that this function has the best behaviour for the first 300 terms after manual validation.
- GaleXtract and TermoStat employ popular association measures: $\chi^2$ (Nagao *et al.* 1976), log likelihood (Dunning 1994), mutual information (Church and

[14] http://tubo.lirmm.fr/biotex/

[15] http://gramatica.usc.es/~gamallo/php/gale-extra/gale-extra2.1/index.php

[16] http://www.nactem.ac.uk/software/termine/

[17] http://termostat.ling.umontreal.ca

Hanks 1990) and symmetric conditional probability (Silva and Lopes 1999) in the former, and $\chi^2$, log likelihood and log odds ratio (Everitt 1992) in the latter. Termine makes use of C-value (Frantzi *et al.* 2000). BioTex employs LIDF-value, F-OCapi and F-TFIDF-C, where the two latter combine C-value with Okapi and TFIDF respectively (Lossio-Ventura, Jonquet, Roche and Teisseire 2014b, 2014c).

The evaluation focused on open-access systems with a GUI that could be used to support terminology and terminography research. The software that did not contribute with different metrics was ignored; for example, AntConc[18] calculates keyness values with $\chi^2$ or log likelihood, which are already included in TermoStat. Moreover, the software that did not allow a proper comparative evaluation was not taken into consideration; for example, Translated-Labs Term Extractor[19] only returns twenty term candidates. In fact, although there are a number of so-called term-extraction programs on the Web, most of them (e.g. Anchovy[20] or Okapi's Rainbow)[21] should be regarded as no more than tools that return frequency-sorted lists of ngrams.

DEXTER was evaluated by assessing the precision of the top-ranked 200 unigrams, bigrams and trigrams extracted as term candidates from two corpora of different domains and languages. The following experiments were devised to demonstrate that SRC can outperform not only the metrics of other ATE systems but also the results obtained by the single metrics of S (Salience), R (Relevance) and C (Cohesion). On the one hand, term extraction was performed on our sample corpus, that is, the corpus of 200 English-written documents about electronics. Tables 8, 9 and 10

[18] http://www.laurenceanthony.net/software/antconc/

[19] http://labs.translated.net/terminology-extraction/

[20] http://www.maxprograms.com/products/anchovy.html

[21] http://okapi.sourceforge.net

present the results of precision after manual validation of the candidates extracted from this English corpus.

| # candidates | SRC | S | R | $\chi^2$ | Log likelihood | Log odds ratio | F-OCapi | F-TFIDF-C | LIDF-value |
|---|---|---|---|---|---|---|---|---|---|
| 1-50 | 0.92 | 0.92 | 0.60 | 0.82 | 0.80 | 0.84 | 0.82 | 0.84 | 0.78 |
| 51-100 | 0.72 | 0.72 | 0.68 | 0.66 | 0.58 | 0.68 | 0.56 | 0.62 | 0.50 |
| 101-150 | 0.68 | 0.68 | 0.40 | 0.66 | 0.52 | 0.58 | 0.66 | 0.54 | 0.44 |
| 151-200 | 0.72 | 0.72 | 0.48 | 0.64 | 0.66 | 0.62 | 0.48 | 0.48 | 0.32 |
| [1-200] | 0.76 | 0.76 | 0.54 | 0.70 | 0.64 | 0.68 | 0.63 | 0.62 | 0.51 |

Table 8. Precision with unigrams (English corpus).

| # candidates | SRC | S | R | C | $\chi^2$ | Log likelihood |
|---|---|---|---|---|---|---|
| 1-50 | 0.86 | 0.86 | 0.50 | 0.76 | 0.60 | 0.66 |
| 51-100 | 0.82 | 0.82 | 0.42 | 0.76 | 0.56 | 0.44 |
| 101-150 | 0.66 | 0.66 | 0.40 | 0.46 | 0.44 | 0.46 |
| 151-200 | 0.68 | 0.68 | 0.52 | 0.66 | 0.34 | 0.40 |

| | | | | | | |
|---|---|---|---|---|---|---|
| [1-200] | 0.76 | 0.76 | 0.46 | 0.66 | 0.49 | 0.49 |

| Log odds ratio | Mutual information | SCP | C-value | F-OCapi | F-TFIDF-C | LIDF-value |
|---|---|---|---|---|---|---|
| 0.58 | 0.24 | 0.28 | 0.78 | 0.64 | 0.62 | 0.72 |
| 0.52 | 0.28 | 0.38 | 0.70 | 0.46 | 0.56 | 0.50 |
| 0.44 | 0.34 | 0.34 | 0.66 | 0.68 | 0.50 | 0.40 |
| 0.40 | 0.42 | 0.38 | 0.60 | 0.56 | 0.58 | 0.44 |
| 0.48 | 0.32 | 0.35 | 0.69 | 0.58 | 0.56 | 0.51 |

Table 9. Precision with bigrams (English corpus).

| # candidates | SRC | S | R | C | $\chi^2$ | Log likelihood |
|---|---|---|---|---|---|---|
| 1-50 | 0.90 | 0.90 | 0.66 | 0.90 | 0.52 | 0.56 |
| 51-100 | 0.84 | 0.74 | 0.70 | 0.80 | 0.50 | 0.44 |
| 101-150 | 0.56 | 0.56 | 0.50 | 0.56 | 0.26 | 0.34 |
| 151-200 | 0.70 | 0.56 | 0.62 | 0.68 | 0.36 | 0.34 |
| [1-200] | 0.75 | 0.69 | 0.62 | 0.74 | 0.41 | 0.42 |

| Log odds ratio | Mutual information | SCP | C-value | F-OCapi | F-TFIDF-C | LIDF-value |
|---|---|---|---|---|---|---|
| 0.44 | 0.48 | 0.50 | 0.86 | 0.64 | 0.64 | 0.78 |
| 0.42 | 0.44 | 0.40 | 0.68 | 0.42 | 0.42 | 0.60 |
| 0.32 | 0.46 | 0.54 | 0.64 | 0.24 | 0.34 | 0.34 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 0.32 | 0.40 | 0.32 | 0.66 | 0.50 | 0.52 | 0.50 |
| 0.37 | 0.45 | 0.44 | 0.71 | 0.45 | 0.48 | 0.55 |

Table 10. Precision with trigrams (English corpus).

On the other hand, term extraction was also performed on a corpus of 100 Spanish texts (273,476 tokens) about odontostomatology, whose true domains were Health [2841], Health care profession [2841001], Health policy [2841002], Illness [2841003], Medical science [2841004], Nutrition [2841005], Pharmaceutical industry [2841006] and Life sciences [3606003]; the false domains were Science [36], Natural and applied sciences [3606] and Applied sciences [3606001], since their terms are commonly found in many scientific disciplines. The documents were obtained from the scientific journal *Avances en Odontoestomatología*.[22] In this case, additional preprocessing was required during corpus compilation, where the English abstract and the list of bibliographical references were removed in each document. DEXTER extracted 2,642 unigrams, 385 bigrams and 110 trigrams as term candidates. Tables 11, 12 and 13 present the results of precision after manual validation of the candidates extracted from this Spanish corpus.

| # candidates | SRC | S | R | $\chi^2$ | Log likelihood | Log odds ratio |
|---|---|---|---|---|---|---|
| 1-50 | 0.98 | 0.98 | 0.96 | 0.68 | 0.58 | 0.92 |
| 51-100 | 0.88 | 0.82 | 0.82 | 0.76 | 0.58 | 0.84 |
| 101-150 | 0.86 | 0.78 | 0.82 | 0.66 | 0.54 | 0.78 |
| 151-200 | 0.78 | 0.70 | 0.74 | 0.58 | 0.54 | 0.72 |
| [1-200] | 0.88 | 0.82 | 0.84 | 0.67 | 0.56 | 0.82 |

[22] http://scielo.isciii.es/scielo.php?script=sci_serial&pid=0213-1285&lng=es&nrm=iso

| F-OCapi | F-TFIDF-C | LIDF-value |
|---|---|---|
| 0.84 | 0.78 | 0.68 |
| 0.72 | 0.72 | 0.62 |
| 0.64 | 0.68 | 0.56 |
| 0.46 | 0.64 | 0.60 |
| 0.67 | 0.71 | 0.62 |

Table 11. Precision with unigrams (Spanish corpus).

| # candidates | SRC | S | R | C | $\chi^2$ | Log likelihood | Log odds ratio | Mutual information | SCP | C-value | F-OCapi | F-TFIDF-C | LIDF-value |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1-50 | 0.90 | 0.90 | 0.76 | 0.68 | 0.42 | 0.44 | 0.46 | 0.26 | 0.34 | 0.40 | 0.52 | 0.56 | 0.48 |
| 51-100 | 0.86 | 0.86 | 0.82 | 0.74 | 0.36 | 0.34 | 0.36 | 0.26 | 0.28 | 0.38 | 0.40 | 0.40 | 0.44 |
| 101-150 | 0.78 | 0.74 | 0.78 | 0.66 | 0.40 | 0.30 | 0.28 | 0.38 | 0.28 | 0.42 | 0.32 | 0.36 | 0.40 |
| 151-200 | 0.74 | 0.74 | 0.72 | 0.70 | 0.18 | 0.32 | 0.36 | 0.26 | 0.34 | 0.40 | 0.44 | 0.42 | 0.48 |
| [1-200] | 0.82 | 0.81 | 0.77 | 0.70 | 0.34 | 0.35 | 0.37 | 0.29 | 0.31 | 0.40 | 0.42 | 0.44 | 0.45 |

Table 12. Precision with bigrams (Spanish corpus).

| # candidates | SRC | S | R | C | $\chi^2$ | Log likelihood | Log odds ratio | Mutual information | SCP | C-value | F-OCapi | F-TFIDF-C | LIDF-value |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1-50 | 0.82 | 0.72 | 0.76 | 0.82 | 0.52 | 0.54 | 0.48 | 0.44 | 0.50 | 0.54 | 0.56 | 0.50 | 0.54 |
| 51-100 | 0.68 | 0.68 | 0.64 | 0.64 | 0.34 | 0.26 | 0.20 | 0.32 | 0.32 | 0.22 | 0.20 | 0.24 | 0.26 |
| [1-100] | 0.75 | 0.70 | 0.70 | 0.73 | 0.43 | 0.40 | 0.34 | 0.38 | 0.41 | 0.38 | 0.38 | 0.37 | 0.40 |

Table 13. Precision with trigrams (Spanish corpus).[23]

Tables 14 and 15 display the values of each SRC coefficient for the unigrams, bigrams and trigrams in the two corpora.

| type | α | β | γ |
|---|---|---|---|
| unigrams | 1 | 0 | - |
| bigrams | 1 | 0 | 0 |
| trigrams | 0.2 | 0 | 0.8 |

Table 14. SRC coefficients for the English corpus.

| type | α | β | γ |
|---|---|---|---|
| unigrams | 0.8 | 0.2 | - |

[23] Due to the limited number of trigrams extracted from the Spanish corpus by DEXTER, only the top-ranked 100 candidates were taken into account in the evaluation of precision.

| | | | |
|---|---|---|---|
| bigrams | 0.6 | 0.4 | 0 |
| trigrams | 0.6 | 0 | 0.4 |

Table 15. SRC coefficients for the Spanish corpus.

The evaluation of precision required manual validation, since term recognition is not a fully reliable process. IATE is not a full-fledged terminological database, not to mention duplicates, incomplete entries, misspellings or obsolete data that can be found (Zorrilla-Agut 2013). The key issue is that gold standards in terminology can be recognised as exemplars of quality but not of perfection. Indeed, term recognition with IATE raised both false negatives and false positives; for example, manual evaluation revealed 10 unigrams (e.g. *amp* or *watt*), 48 bigrams (e.g. *Darlington transistor* or *Schmitt inverter*) and 127 trigrams (e.g. *Kirchoffs Current Law* or *Wheatstone Bridge circuit*) as false negatives and 37 unigrams (e.g. *device, maximum* or *unit*) and 6 bigrams (e.g. *input signal*) as false positives among the top 200 SRC-ranked ngrams from the English corpus. In both experiments, manual validation was carried out by three terminologists. The problem is that, "since the definition of *termhood* is pretty vague, it is likely that experts produce different validations, based on their own intuition of term" (Pazienza *et al.* 2005: 265). Evaluators need to rely on a clear notion of what a term is. A definition of term such as "a designation consisting of one or more words representing a general concept in a special language in a specific subject field" (ISO 704 2009: 34) is very vague for practical purposes. In order to deal effectively with this problem, it should be recalled that terms can be categorized into neonyms, existing forms or translingual loans. For obvious reasons, neonyms and translingual loans don’t raise any problem during term validation. However, existing forms do not help to determine clear criteria to identify domain-specific terms. Paradoxically, most ATE researchers (cf. Park *et al.*

2002; Zhang *et al.* 2008; Knoth *et al.* 2009) are concerned with the number and/or profile of the evaluators involved in the experiments rather than with a clear definition of what a term is. All experts agree that defining such a criterion is not an easy task, so how can evaluators decide that a given candidate is a “good glossary item” or it is “characteristic for the domain”? In the evaluation of SRC, we crystallized this decision-making process into a flowchart (Figure 4), where the most critical question was that of the relevance of definitions.

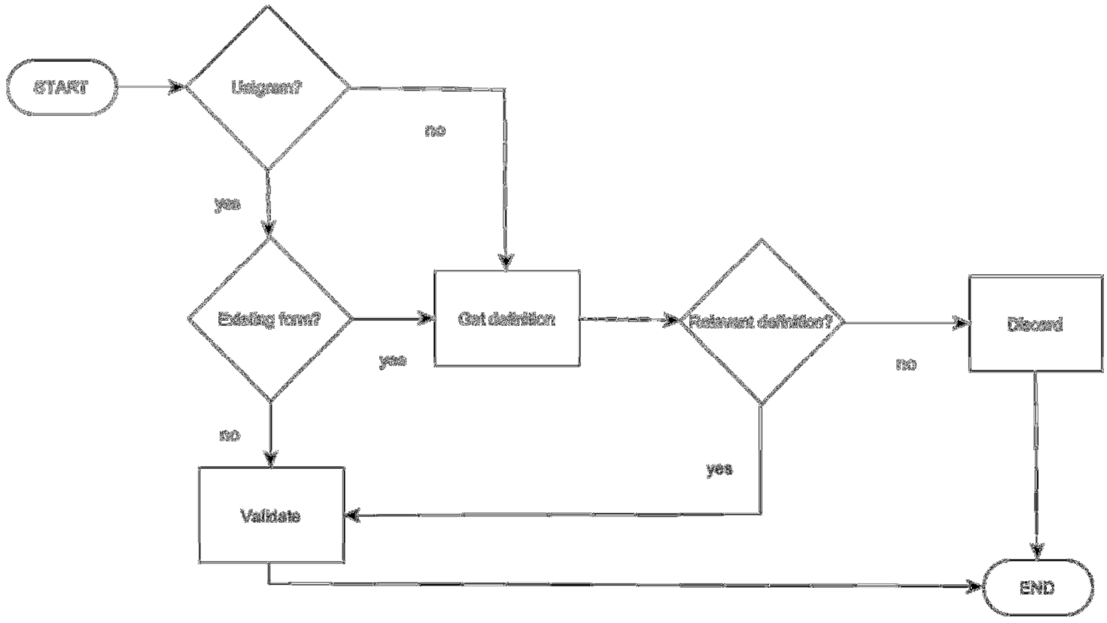


Figure 5. Flowchart for term validation.

Thus, once the definition of the term candidate was obtained from the corpus itself or from other specialized resources (e.g. Google Books and Wikipedia), the evaluator explored the *definiens* to find at least one lexical item that could be related to any of the true domains of the corpus; if so, the ngram was validated. As scientific and technical definitions are aimed at giving insights into the subject matter to which the defined term pertains, it can be concluded that the *definiens* provides a means for the discovery of the specialized domain of the *definiendum.* Thus, the flowchart was used as a guideline to find verifiable evidence for both single- and multi-word terms. This model of evaluation favours final agreement among human judgements, since decisions must be based on textual evidence.

The flowchart proved to be particularly useful to differentiate between complex terminological units and phraseological collocations. For example, should *coil of wire* and *coil rotation* be considered terms of electronics? On the one hand, *coil of wire* was selected as a term because its definition includes three words, i.e. *resistor, voltage* and *inductor*, which undoubtedly pertain to the electronics terminology:

> A coil of wire is simply a resistor as far as steady voltage is concerned, but for alternating voltages it behaves as an inductor. (Sinclair 2011: 36)

On the other hand, the flowchart did not reveal *coil rotation* as a term but only as a statistically significant co-occurrence (i.e. collocation). To illustrate, we present some valid and non-valid term candidates related to the domain of the English corpus:

- Valid term candidates: (a) unigrams: *capacitance, farad, galvanometer* or *voltage*; (b) bigrams: *Ohm's law, PN junction, RLC circuit* or *square wave*; and (c) trigrams: *magnetic flux linkage, metal film resistor, solid state relay* or *voltage divider circuit.*
- Non-valid term candidates: (a) unigrams: *centimeter, theorem, tutorial* or *vector*; (b) bigrams: *capacitance value, high frequency, input signal* or *time constant*; and (c) trigrams: *changes in temperature, Greek symbol phi, speed of rotation* or *upper threshold level.*

The analysis of non-valid term candidates revealed that false positives usually take the form of (a) common words, e.g. *maximum* or *value*, and (b) complex ngrams nested in longer multi-word candidates, e.g. *width modulation* (*pulse width modulation*) or *permanent magnet DC* (*permanent magnet DC motor*).

Moreover, although it is well known that nouns and noun phrases make up the bulk of term candidates, both experiments showed that some adjectives (e.g. *astable, capacitive, inductive*, *sinusoidal* or *voltaic* [electronics]; *birradicular, dentinario, estomatológico, gingival, hemostático, malar, mesiodistal* or *periodontal* [odontostomatology]) and verbs (e.g. *amplify, rectify* or *regulate* [electronics]; *bruñir* or *suturar* [odontostomatology]) can be closely linked to specialized domains.

Both experiments demonstrated that the best precision was obtained with SRC. Indeed, if ranges are inspected, it can be realized that on just one occasion SRC was outperformed in the twenty-two different cut-off points along the top 200 unigrams, bigrams and trigrams. Moreover, like C-value (Termine) and the metrics based on C-value (BioTex), SRC shows a rather gradual distribution of positive candidates in the extracted list, where true terms tend to be attracted to the top of the list.

Since the values of the SRC coefficients can be different for unigrams, bigrams and trigrams (Tables 14 and 15), it makes sense to have separated lists of term candidates to evaluate the behaviour of the various types of ngrams, providing useful evidence to support corpus-based terminology and terminography research. If we finally choose to compile a global list of candidates, then the challenge is to significantly correlate scores that belong to different normal distributions (i.e. each with a different mean and standard deviation). In this context, Z-score standardization helps to determine which unigrams, bigrams and trigrams are at a similar distance from the mean in their respective distributions, so we can position candidates with similar Z-scores close together in the ranking. To illustrate, the global list of ngrams from the English corpus showed the best precision with the top-ranked 200 ngrams (0.88); in this case, precision was also better in each range: 0.94 for 1-50, 0.98 for 51-100, 0.84 for 101-150 and 0.76 for 151-200.

## 4. *Conclusions*

This article analyzed the term-extraction process that takes place in DEXTER, an online workbench that consists of a suite of corpus and terminological tools with such diverse functionalities as corpus compilation and management, document indexation and retrieval, query elaboration, textual exploration, and term weighting and validation. The components of DEXTER that play a critical role in the ATE process, and that have contributed to make some headway in the field of ATE, are (i) the stopword-detection method, (ii) the SRC metric, and (iii) the IATE database. On the one hand, DEXTER is provided with a multilingual and multi-domain method that aims to reduce the inventory of term candidates by filtering simple and complex ngrams on the basis of the common stopwords detected automatically in the given specialized corpus. On the other hand, SRC is a parameterized metric for term ranking that relies on the theoretical principles of (a) salience, which measures the prevalence of terms in the document collection, (b) relevance, which measures the tendency in the usage of terms between a domain-specific corpus and a general-purpose one, and (c) cohesion, which measures the degree of stability of multi-word terms. Finally, the IATE database not only enables DEXTER to recognize domain-specific terms but, even more important, helps to determine the values of the SRC coefficients by placing best term candidates in the foreground.

This research was able to demonstrate that the requirements described in the introduction were successfully implemented in DEXTER:

Requirement 1. SRC outperforms the results of well-known statistical-significance metrics (e.g. $\chi^2$, log likelihood, log odds ratio, mutual information and SCP) as well as C-value and other metrics based on the latter (e.g. F-OCapi and F-TFIDF-C). The effectiveness of SRC, whose parameters enable the system to deal with a wide diversity of specialized corpora, was measured with respect to precision.

Requirement 2. DEXTER can extract simple and complex terms (i.e. unigrams, bigrams and trigrams) with the same metric.

Requirement 3. DEXTER can recognize domain-specific terms of different grammatical categories, e.g. nouns, verbs and adjectives, by means of shallow lexical filters rather than elaborate morphosyntactic patterns. It has been proved that the precision of DEXTER is better than that of those systems that adopt a hybrid method for term extraction (e.g. BioTex, GaleXtract, Termine and TermoStat), where the statistical approach is applied to the output of POS taggers.

Requirement 4. DEXTER is capable of processing corpora in several languages (i.e. English, French, Italian and Spanish) by means of knowledge-poor procedures that do not entail sophisticated NLP techniques. Indeed, this term extractor is provided with a few language-dependent resources—i.e. stemmer, lemmatizer and stopword list. For each language, the stopword list contains (a) functional words that were manually obtained from the grammar of the language, and (b) common words that are discovered at runtime using an adaptive method applied to the domain-specific corpus and a general corpus from the Leipzig Corpora Collection. Together with the IATE database, these resources serve to automatically tune the SRC coefficients for the extraction of specialized lexical units.

Requirement 5. DEXTER integrates corpus management and term extraction in a single platform with a user-friendly interface primarily intended for linguistic research.

Future research is aimed at increasing the number of languages (currently four) and the types of ngrams (currently up to trigrams) that DEXTER can process.

### *Acknowledgements*

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

### *References*

Ahmad, K., Gillam, L., and Tostevin, L. 2000. Weirdness indexing for logical document extrapolation and retrieval (WILDER). In Voorhees, E. M., and Harman, D. K. (eds.), *Proceedings of the 8th Text Retrieval Conference*. Washington: National Institute of Standards and Technology, pp. 717-724.

Ahrenberg, L. 2009. Term extraction: A review. Retrieved from http://www.ida.liu.se/~lah/Publications/tereview_v2.pdf

Alajmi, A., Saad, E. M., and Darwish, R. R. 2012. Toward an ARABIC stop-words list generation. *International Journal of Computer Applications* 46 (8): 8-13.

Asubiaro, T. V. 2013. Entropy-based generic stopwords list for Yoruba texts. *International Journal of Computer and Information Technology* 2 (5): 1065-1068.

Barcala, M., Domínguez-Noya, E., Gamallo, P., López, M., Moscoso, E., Rojo, G., Santalla, P., and Sotelo, S. 2007. A corpus and lexical resources for multi-word terminology extraction in the field of economy. In *Proceedings of the 3rd Language and Technology Conference*, Poznan, pp. 355-359.

Biemann, C., Heyer, G., Quasthoff, U., and Richter, M. 2007. The Leipzig Corpora Collection: monolingual corpora of standard size. In *Proceedings of Corpus Linguistic 2007*, Birmingham.

Brants, T. 2004. Natural language processing in information retrieval. In *Proceedings of the 14th Meeting of Computational Linguistics*, Antwerp, pp. 1-13.

Church, K.W., Gale, W., Hanks, P., and Hindle, D. 1991. Using statistics in lexical analysis. In Zernik, U. (ed.), *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*. Hillsdale: Lawrence Erlbaum Associates, pp. 115-164.

Church, K.W., and Hanks, P. 1990. Word association norms, mutual information and lexicography. *Computational Linguistics* 6 (1): 22-29.

Conde, A., Larrañaga, M., Arruarte, A., Elorriaga, J. A., and Roth, D. 2016. LiteWi: a combined term extraction method for eliciting educational ontologies from textbooks. *Journal of the Association for Information Science and Technology* 67 (2): 380-399.

Conrado, M. S., Felippo, A., Pardo, T. A. S., and Rezende, S. O. 2014. A survey of automatic term extraction for Brazilian Portuguese. *Journal of the Brazilian Computer Society* 20 (12): 1-28.

Deane, P. 2005. A nonparametric method for extraction of candidate phrasal terms. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*. Michigan: Association for Computer Linguistics, pp. 605-613.

Drouin, P. 2003. Term extraction using non-technical corpora as a point of leverage. *Terminology* 9 (1): 99-117.

Dunning, T. 1994. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* 19 (1): 61-74.

Everitt, B. 1992. *The Analysis of Contingency Tables*. London: Chapman & Hall/CRC.

Fedorenko, D., Astrakhantsev, N., and Turdakov, D. 2013. Automatic recognition of domain-specific terms: an experimental evaluation. In *Proceedings of the 9th Spring Researcher's Colloquium on Database and Information Systems*, pp. 15-23.

Fox, C. 1990. A stop list for general text. *ACM-SIGIR Forum* 24: 19-35.

Francis, W. N., and Kučera, H. 1982. *Frequency Analysis of English Usage: Lexicon and Grammar*. Boston: Houghton Mifflin.

Frantzi, K., and Ananiadou, S. 1996. Extracting nested collocations. In *Proceedings of the 16th International Conference on Computational Linguistics*. Morristown: Association for Computational Linguistics, pp. 41-46.

Frantzi, K., Ananiadou, S., and Mima, H. 2000. Automatic recognition of multi-word terms. *International Journal of Digital Libraries* 3 (2): 117-132.

Gale, W., and Church, K.W. 1991. Concordances for parallel texts. In *Proceedings of the 7th Annual Conference of the UW Center for the New OED and Text Research*, Oxford, pp. 40-62.

Haan, P. 1992. The optimum corpus sample size? In Leitner, G. (ed.), *New Dimensions in English Language Corpora*. Berlin-NewYork: Mouton de Gruyter, pp. 3-19.

Harman, D. 1986. An experimental study of factors important in document ranking. In *Proceedings of the 9th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Pisa, pp. 186-193.

Hatcher, E., Gospodnetic, O., and McCandless, M. 2010. *Lucene in Action*. Greenwich: Manning.

Hunston, S. 2008. Collection strategies and design decisions. In Lüdeling, A., and Kytö, M. (eds.), *Corpus Linguistics: An International Handbook*. Volume 1. Berlin-New York: Mouton de Gruyter, pp. 154-168.

ISO 704. 2009. *Terminology Work – Principles and Methods*. Geneva: International Organization for Standardization.

Ittoo, A., Maruster, L., Wortmann, H., and Bouma, G. 2010. Textractor: a framework for extracting relevant domain concepts from irregular corporate textual datasets. In Abramowicz, W., and Tolksdorf, R. (eds.), *Business Information Systems*. Lecture Notes in Business Information Processing, vol. 47. Heidelberg: Springer, pp. 71-82.

Jacquey, E., Tutin, A., Kister, L., Jacques, M., Hatier, S., and Ollinger, S. 2013. Filtrage terminologique par le lexique transdisciplinaire scientifique: une expérimentation en sciences humaines. In *Proceedings of the 10th International Conference on Terminology and Artificial Intelligence (TIA 2013)*. Villetaneuse, pp. 121-128.

Justeson, J. S., and Katz, S. M. 1995. Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering* 1 (1): 9-27.

Kageura, K., and Umino, B. 1996. Methods of automatic term recognition: A review. *Terminology* 3 (2): 259-289.

Karystianis, G., Buchan, I., and Nenadic, G. 2014. Mining characteristics of epidemiological studies from Medline: a case study in obesity. *Journal of Biomedical Semantics* 5, 22: 1-11.

Khosrow-Pour, M. 2009. *Encyclopedia of Information Science and Technology*. Hershey: Information Science Reference.

Knoth, P., Schmidt, M., Smrz, P., and Zdráhal, Z. 2009. Towards a framework for comparing automatic term recognition methods. In *Proceedings of the 8th*

*Annual Conference Znalosti*. Bratislava: Informatics and Information Technology STU, pp. 83-94.

Koester, A. 2010. Building small specialized corpora. In O'Keeffe, A., and McCarthy, M. (eds.), *The Routledge Handbook of Corpus Linguistics*. London: Routledge, pp. 66-79.

Korkontzelos, I., Klapaftis, I., and Manandhar, S. 2008. Reviewing and evaluating automatic term recognition techniques. In *Proceedings of the 6th International Conference on Advances in Natural Language Processing*. Berlin-Heidelberg: Springer, pp. 248-259.

Lochbaum, K. E., and Streeter, L. A. 1989. Comparing and combining the effectiveness of latent semantic indexing and the ordinary vector space model for information retrieval. *Information Processing and Management* 25 (6): 665-676.

Lossio-Ventura, J.A., Jonquet, C., Roche, M., and Teisseire M. 2014a. BioTex: a system for biomedical terminology extraction, ranking and validation. In *Proceedings of the 13th International Semantic Web Conference*, pp. 157-160.

Lossio-Ventura, J.A., Jonquet, C., Roche, M., and Teisseire M. 2014b. Towards a mixed approach to extract biomedical terms from text corpus. *International Journal of Knowledge Discovery in Bioinformatics* 4 (1): 1-15.

Lossio-Ventura, J.A., Jonquet, C., Roche, M., and Teisseire M. 2014c. Yet another ranking function to automatic multi-word term extraction. In *Proceedings of the 9th International Conference on Natural Language Processing*, Warsaw.

Luhn, H. P. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development* 2 (2): 159-165.

Marín, M.J. 2015. Measuring precision in legal term mining: a corpus-based validation of single and multi-word term recognition methods. *ESP World* 46: 1-23.

Merkel, M., Foo, J., and Ahrenberg, L. 2013. IPhraxtor - a linguistically informed system for extraction of term candidates. In *Proceedings of the 19th Nordic Conference on Computational Linguistics*. Oslo: Linkoping University Electronic Press, pp. 121-132.

Meyers, A., He, Y., Glass, Z, and Babko-Malaya, O. 2015. The Termolator: terminology recognition based on chunking, statistical and search-based scores. In *Proceedings of the First Workshop on Mining Scientific Papers: Computational Linguistics and Bibliometrics*, Istanbul, pp. 34-43.

Nagao, M., Mizutani, M., and Ikeda, H. 1976. An automated method of the extraction of important words from Japanese scientific documents. *Transactions of the Information Processing Society of Japan* 17 (2): 110-117.

Oakes, M. 1998. *Statistics for Corpus Linguistics*. Edinburgh: Edinburgh University Press.

Park, Y., Byrd, R. J., and Boguraev, B. 2002. Automatic glossary extraction: beyond terminology identification. In *Proceedings of the 19th International Conference on Computational Linguistics*. Taipei: Howard International House and Academia Sinica, pp. 1-7.

Paulo, J. L., and Mamede, N. J. 2004. Terms spotting with linguistics and statistics. In De Ita Luna, G., Fuentes Chávez, O., and Osorio Galindo, M. (eds.), *Proceedings of the International Workshop Taller de Herramientas y Recursos Linguísticos para el Español y el Portugués, IX Iberoamerican Conference on Artificial Intelligence*, pp. 298-304.

Pazienza, M. T., Pennacchiotti, M., and Zanzotto, F. M. 2005. Terminology extraction: an analysis of linguistic and statistical approaches. In Sirmakessis, S. (ed.), *Knowledge Mining*. Studies in Fuzziness and Soft Computing, vol. 185. Heidelberg: Springer, pp. 255-279.

Periñán-Pascual, C. 2015. The underpinnings of a composite measure for automatic term extraction: the case of SRC. *Terminology* 21 (2): 151-179.

Quasthoff, U., Richter, M., and Biemann, C. 2006. Corpus portal for search in monolingual corpora. In *Proceedings of LREC-06*, Genova, pp. 1799-1802.

Robertson, S.E., Walker, S., and Beaulieu, M. 1998. Okapi at TREC-7: Automatic ad hoc, filtering, VLC and interactive track. In *Proceedings of the 7th Text Retrieval Conference*, Gaithersburg: National Institute of Standards and Technology, pp. 253-264.

Sajjacholapunt, P., and Joy, M. 2015. Analysing features of lecture slides and past exam paper materials. Towards automatic associating E-materials for self-revision. In *Proceedings of the 7th International Conference on Computer Supported Education*, Lisbon: SciTePress, pp. 169-176.

Salton, G. (ed.) 1971. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Englewood Cliffs: Prentice-Hall.

Salton, G., and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management* 24 (5): 513-523.

Salton, G., and McGill, M. 1983. *Introduction to Modern Information Retrieval.* New York: McGraw Hill.

Salton, G., Wong, A., and Yang, C. S. 1975. A vector space model for automatic indexing. *Communications of the ACM* 18 (11): 613-620.

Salton, G., Yang, C. S., and Yu, C. T. 1975. A theory of term importance in automatic text analysis. *Journal of the American Society for Information Science* 26 (1): 33-44.

Silva, J.F., Dias, G., Guilloré, S., and Lopes, G.P. 1999. Using LocalMaxs algorithm for the extraction of contiguous and non-contiguous multiword lexical units. In Barahona, P. (ed.), *Progress in Artificial Intelligence: 9th Portuguese Conference on AI*. Heidelberg: Springer, pp. 113-132.

Silva, J.F., and Lopes, G.P. 1999. A local maxima method and a fair dispersion normalization for extracting multiword units. In *Proceedings of the 6th Meeting on the Mathematics of Language*, Orlando, pp. 369-381.

Sinclair, I. 2011. *Electronics Simplified*. Oxford: Newnes-Elsewier.

Singhal, A., Buckley, C., and Mitra, M. 1996. Pivoted document length normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM press, pp. 21-29.

Sinka, M. P., and Corne, D. W. 2003. Towards modernised and web-specific stoplists for web document analysis. In *Proceedings of IEEE Web Intelligence 2003*. Los Alamitos (California): IEEE Computer Society, pp. 396-404.

Smadja, F., McKeown, K.R., and Hatzivassiloglou, V. 1996. Translating collocations for bilingual lexicons: a statistical approach. *Journal of Computational Linguistics* 22 (1): 1-38.

Sun, Q., Shaw, D., and Davis, C. H. 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.

Thurmair, G. 2003. Making term extraction tools usable. In *Proceedings of The Joint Conference of the 8th International Workshop of the European Association of*

*Machine Translation and the 4th Controlled Language Applications Workshop*. Dublin: European Association for Machine Translation, pp. 1-10.

Vivaldi, J., Màrquez, L., and Rodríguez, H. 2001. Improving term extraction by system combination using boosting. In *Proceedings of the 12th European Conference on Machine Learning*. Heidelberg: Springer, pp. 515-526.

Vivaldi, J., and Rodríguez, H. 2007. Evaluation of terms and term extraction systems: a practical approach. *Terminology* 13 (2): 225-248.

Wermter, J., and Hahn, U. 2005. Finding new terminology in very large corpora. In Clark, P., and Schreiber, G. (eds.), *Proceedings of the 3rd International Conference on Knowledge Capture*. Alberta: Association for Computing Machinery, pp. 137-144.

Wiechmann, D., and Fuhs, S. 2006. Corpus linguistics resources. Concordancing software. *Corpus Linguistics and Linguistic Theory* 2 (1): 109-30.

Wong, W., Liu, W., and Bennamoun, M. 2008. Determination of unithood and termhood for term recognition. In Song, M., and Wu, Y. (eds.), *Handbook of research on text and web mining technologies*. Hershey-New York: IGI Global, pp. 500-529.

Zadeh, B. Q., and Handschuh, S. 2014a. Evaluation of technology term recognition with random indexing. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*. Reykjavik: European Language Resources Association, pp. 4027-4032.

Zadeh, B. Q., and Handschuh, S. 2014b. The ACL RD-TEC: a dataset for benchmarking terminology extraction and classification in computational linguistics. In *Proceedings of the 4th International Workshop on Computational Terminology*, Dublin: Association for Computational Linguistics, pp. 52-63.

Zhang, Z., Iria, J., Brewster, C., and Ciravegna, F. 2008. A comparative evaluation of term recognition algorithms. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*. Luxemburg: European Language Resources Association, pp. 2108-2113.

Zorrilla-Agut, P. 2014. When IATE met LISE: LISE clean-up and consolidation tools take on the IATE challenge. In Budin, G., and V. Lušicky (eds.), *Languages for Special Purposes in a Multilingual, Transcultural World. Proceedings of the 19th European Symposium on Languages for Special Purposes*. Vienna: University of Vienna, pp. 536-545.

Zou, F., Wang, F. L., Deng, X., Han, S., and Wang, L. S. 2006. Automatic construction of Chinese stop word list. In *Proceedings of the 5th WSEAS International Conference on Applied Computer Science*, Hangzhou, pp. 1010-1015.