Automatic Matching and Expansion of Abbreviated Phrases without Context

Chloé Artaud[1,2], Antoine Doucet[1],
Vincent Poulain D'Andecy[3], and Jean-Marc Ogier[1]

[1] L3i, University of La Rochelle, France,
{chloe.artaud, antoine.doucet, jean-marc.ogier}@univ-lr.fr,
[2] Direction Générale de l'Armement, France,
[3] Yooz, Aimargues, France,
Vincent.PoulainDAndecy@yooz.fr,

Abstract. In many documents, like receipts or invoices, textual information is constrained by the space and organization of the document. The document information has no natural language context, and expressions are often abbreviated to respect the graphical layout, both at word level and phrase level. In order to analyze the semantic content of these types of document, we need to understand each phrase, and particularly each name of sold products. In this paper, we propose an approach to find the right expansion of abbreviations and acronyms, without context. First, we extract information about sold products from our receipts corpus and we analyze the different linguistic processes of abbreviation. Then, we retrieve a list of expanded names of products sold by the company that emitted receipts, and we propose an algorithm to pair extracted names of products with the corresponding expansions. We provide the research community with a unique document collection for abbreviation expansion.

Keywords: abbreviations, string distance, information extraction

1 Introduction

In the general objective of detecting false documents by verifying the information contained in the document, we must extract and model this information in order to be able to compare it with information that we will search for in external resources. However, administrative documents, such as payslips, administrative forms, invoices, proofs of purchase, have many constraints, particularly with regard to the document layout and the space available for each bit of information. Indeed, administrative documents, except for letters and contracts, are intended to be sufficiently concise and clear to be analyzed at first glance by humans.

This conciseness is often expressed by a layout containing precise tables and locations for the different elements specific to each type of document. For instance, in an invoice, we will often find a logo in the upper part, a header containing metadata, a table with names of products or services on the left, prices on the right, a total amount below, and the fine print at the end of the document. This highly organized structure imposes spatial constraints on textual and graphical elements: all the information necessary for the usefulness of the document must fit into a single document, often a single page for practical reasons. While logos and other graphic elements can be reduced, the textual information must be shortened while remaining sufficient to be understandable.

This constraint, common to many document types, allows us to observe the linguistic phenomenon of reduction. These abbreviations, that appear in many forms, are a bottleneck for the automatic analysis of document information. Indeed, it is very difficult to extract and interpret information that is not exhaustive and not fully explicit. We must therefore seek to associate abbreviations with their full form, or "expansion".

The task of associating abbreviations with their possible expansions is not simple because it involves many constraints: the absence of natural language context, the diversity of the types of abbreviations, the double level of abbreviation in words and in phrase, the ambiguity of the product names...

In Section 2 of this paper on related work, we show that the context-less automatic matching of abbreviations and their complete forms is a problem that has not been much studied in the state of the art. Section 3 presents our data set and analyses the different type of shortened forms observed. Section 4 explains the proposed approach for pairing abbreviated phrases and their possible expansions. Section 5 describes and discusses the results of this approach. We finally conclude and provide perspectives in Section 6.

2 Related Work

2.1 French Linguistics Definitions

French linguistics grammar does not much describe abbreviations in the French language, their socio-linguistic and semantic uses and values. For example, Riegel et al. only grant two pages and is focused in truncation and initialism only [1]. The use of abbreviations are anecdotal to this work as it only belongs to the scriptural aspect of the language and is specific to each writer. To define the abbreviations, it is the graphic aspect that also prevails for Grevisse and Lits [2], except for the phenomenon of reduction, which is the construction of a new form from another having the same meaning, by removal of a part of the full form. Similarly, acronyms enter the current language by becoming a word, as for instance "AIDS" in English. Grevisse and Lits [2] further classify measurement units as symbols, not abbreviations, noting that units (in metrology or chemistry, for example) have lost their abbreviation value to take a symbolic value and are no longer followed by periods.

While the abbreviations in these two reference works are purely personal graphic practice, for Martinet [3], on the contrary, the abbreviations are used to communicate while providing the least effort. In the case of abbreviations, it is a question of keeping the strict necessary to transmit information and thus saving memory, time and space. The abbreviation is an integral part of the evolution of the language: the more frequently an expression is pronounced, the more likely it is to be abbreviated, by eliminating non-specific elements, by truncation or by initialism.

2.2 Abbreviations Disambiguations

To our knowledge, little work exists in Natural Language Processing on the matching of all types of abbreviations with their expansions. However, we can find many works on the extraction of acronyms (often confused with the term "abbreviations") in texts and their disambiguation according to context. This application is particularly present in Biology and Medicine, where acronyms are very numerous, both in academic articles, especially on MEDLINE[4], and in clinical reports.

These studies [4–6] first attempt to extract abbreviations from the text. Authors use pattern-matching rules for mapping an abbreviation to its full form: acronyms are often capitalized, and sometimes in brackets when they appear for the first time [7]. They then use context-based machine learning methods to choose the right expansion from a number of different dictionaries of acronym-expansions pairs, as explained by Gaudan et al. [8]. These methods achieve very good results, but are not applicable to our data.

2.3 String distance methods

Our data are only strings of abbreviated forms and expansion forms, without context. For this reason, we can only work with string-based methods. The best known metric for calculating a distance between two strings is the Levenshtein distance (also known as the edit-distance) [9]. It gives the smallest number of character changes (deletion, insertion, substitution) to switch from one string to another. The weaker the result, the less differences between strings. Numerous variants have been proposed for this metric, in order to take into account also the transpositions (Damerau-Levenshtein) or only substitutions in strings of the same length (Hamming).

In our problem, the strings are of different lengths and only the inserts are of interest to us, at level of characters in words and at level of words in phrases. For this second step, we can place the problem in set theory, considering the phrase as a set of words, which allows us to use measures such as the Jaccard index or the overlap coefficient. The Jaccard index is a simple similarity coefficient between two sets, defines as the size of the intersection divided by the size of the union. This index is between 0 (nothing in common) and 1 (identical sets).

3 Data Collection

We manually created a corpus of 248 product names as typed on sales receipts from a single store. These product names, constrained to be comprehensible for

[4] Medical Literature Analysis and Retrieval System Online, `https://www.nlm.nih.gov/pubs/factsheets/medline.html`

human while being limited to 20 characters, contains many reductions, both at level of words and at level of phrase, and many other obstacles to matching short and long forms.

3.1 Reductions

Abbreviations. In our receipts corpus, the abbreviation is a graphic phenomenon: the word is written in a short form by deleting some of its letters. We find two types of abbreviation in our corpus:

- Apocope: removal of the end of the word
  Apocope abbreviations may or may not be followed by a period (1a) and (1b). We notice that the period is followed by a space only in one case in our corpus. The rest of the time, it serves to separate a short form of a word from another word, in its short or long form. Our corpus contains a particular apocope type, that is not only a graphic phenomenon, but also phonetic and linguistic phenomenon: the word in its shortened form is so much used that it now exists in everyday language. This is the case of "BIO", which is an abbreviation of "*biologique*" ("from organic farming").
- Syncope: deletion of letters inside the word
  Syncope abbreviations also may or may not be followed by a period. We note that sometimes the first and last letters are kept (2a), but more often it is three representative consonants (2b). In other cases, some vowels are deleted in the word (2c).

These different forms of abbreviations present a particular challenge for their recognition: it seems that there are no systematic rules, or pattern, for the construction of abbreviations in this corpus, apart from the presence of the first letter of the word. Table 1 shows some examples of abbreviations, with the expansion of every word, the phrase expansion, the expansion of the corresponding expression in our corpus of expansions and a translation into English.

Initialisms. The first letters of each word of a compound word or of a brand name composed of several words are assembled. If the initialism is pronounced as a word, we consider it as an acronym. In our corpus, initialisms are sometimes followed by a period, contrary to the typographical rules.

Table 2 shows a few examples of initialisms of our data. We can notice that in the three examples, initialisms take the first letter of the prepositions "de" ("*of*") and "à" ("*to*") contrary to the use.

Symbols. Words are sometimes represented by (or contain) mathematical symbols. We thus frequently find the symbols "+" and "/". These special characters create segmentation problems. Their heterogeneous use does not allow systematic treatment. Indeed, in Table 3, we can see that the slash character can separate words (abbreviated or not) to mean a mixture of ingredients (1), whereas

Table 1. Examples of Abbreviations

Type	Abbreviation	Words expansion, Expansion and *English gloss*
(1a)	**CONF.FIG.PROV.R.**FR	*CONFiture FIGues PROVence Reflet FRance*
		Confiture de figues de Provence Reflets de France
		Fig jam from Provence Reflets de France
(1b)	**TAB CHO** CRF NR **BIO**	*TABlette CHOcolat CaRreFour NoiR BIOlogique*
		Chocolat bio noir 74% cacao Carrefour Bio
		Organic chocolate 74% cocoa Carrefour Bio
(2a)	**PT** BEUR.CHO **LT** NOI	*PetiT BEURre CHOcolat LaiT NOIsette*
		Biscuits petit beurre chocolat lait Petit Ecolier
		Butter cookies with milk chocolate Petit Ecolier
(2b)	160G **BLC PLT** 4TR.F	*160 Grammes BLanC PouLeT 4 TRanches Fines*
		Blanc de poulet Carrefour
		Carrefour Chicken Breast
(2c)	**PT** DEJ.FRUITS **RGES**	*PetiT DEJeuner FRUITS RouGES*
		Biscuits petit déjeuner aux fruits rouges Carrefour
		Breakfast biscuits with red berries Carrefour

Table 2. Examples of Initialisms

Initialisms	Expansion and *English gloss*
2X30CL VELOUT.**PDT.**	*Velouté de Pomme De Terre*
	Cream of potato soup
PT L'EVEQUE **RDF**	*Pont l'Eveque Reflet De France*
	Pont l'Eveque Reflet de France
YAF PANIER FRAM/MU	*Yaourt Aux Fruits Panier de Yoplait framboise/mûre*
	Fruit yoghurt Panier de Yoplait raspberry/blackberry

in (2), the slash intervenes between two words (preposition and noun) of the same unit of meaning. It should be noted that this abbreviation "s/s" may correspond to the bigram "without sugars" but also "without salt" in our corpus. The slash is also observed in its signifying form of mathematical symbol (3). In this last case, we can not define the slash as a separator because the digit-slash-digit set, a mathematical fraction, constitutes a single semantic unit: "half".

Other mathematical symbols are frequently used in cash register abbreviations, such as the plus sign and the letter X, used in place of the multiplication cross. In (4), the sign "+" has a true mathematical value: it is to signify the addition of a numerical quantity to the quantity indicated before the character. On the other hand, in (5), the "+" character is inserted between two (abbreviated) words, denoting a product consisting of two entities. The "+" character represents, in both cases, the combination of two units that are not usually assembled, however, the treatment to be performed is not the same in the two examples. In the first case the character must be interpreted as a word to disambiguate, while in the second, the character must be considered as a separator not to be included in the words framing.

Table 3. Examples of Symbols

Type	Initialisms	Expansion and *English gloss*
(1)	BURGER EMM/CHEDDAR	*Burger emmental **et** cheddar* Burger emmental **and** cheddar
(2)	KRISPROLL.S/S	*Krisprolls **sans sucre*** Krisprolls **without sugar**
(3)	BEURRE BIO 1/2 SEL	*Beurre bio **demi**-sel* **Half**-salt organic butter
(4)	CHOCOLATINE X4 + 1 G	*4 chocolatines **et** une gratuite* 4 chocolate croissant **and** one free
(5)	DUO TERRIN+MOUS.CD	*Duo terrine **et** mousse de canard* Duo terrine **and** duck mousse

Numbers. The mixture of digits and letters is particularly problematic in our data because we can not apply a general rule to treat it. Indeed, there are several cases where numbers and letters are mixed:

- Expressions of quantities, associating a number and the abbreviation of a standard measurement unit (85G : "85 grams")
- Expressions of quantities, associating a number with an abbreviation of a word or a complete word (4FRT : "4 fruits", 4FROMAGES : "4 cheeses")
- Number of units, represented by the letter "X" (here meaning "times") contiguous before or after the number (4X125G : "4 units of 125 grams")
- Abbreviations mathematically symbolized, associating a fraction and letters (1/2ECR : "semi-skimmed")
- Names of products or brands containing numbers and letters, which should not be separated (T45 : "type of flour")
- Absence of spaces between letters and numbers (75 VDP OC RG BIO09 : "organic red wine of year 2009")

3.2 Other problematic features

While the different forms of reduction in the corpus present issues of segmentation and analysis of the text, other characteristics, specific to the format of the document, must be highlighted. Indeed, the word segmentation of the receipts is made difficult by various typographic elements, such as the case used, the presence of periods of abbreviations, the absence of spaces or the concatenation of numbers and units of measure. In rare cases, no character (space, point) can identify the break between two words. In the following example, characters "X" and "G" following a number are lexical units themselves and have to be separated from the abbreviation "DESS.".

Example 1. Product name : 4X100GDESS.PANACHE
Segmentation : 4 X 100 G DESS. PANACHE
English gloss : *four times one hundred grams of mixed dessert cream*

Typographic elements.

Letter case. The characters of this corpus are all capitalized, which does not allow to easily visualize initialisms, as it is the case in most works on acronyms and abbreviations in natural language texts. Nor does it make it possible to distinguish proper names, such as brand names for example, which could have helped in the analysis of documents.

Diacritics. The corpus does not present any diacritic sign, that is, no accent or cedilla. This absence is also a handicap for the analysis because it can cause the ambiguity of certain words, such as PATE, which can correspond to *pâte* ("paste") or *pâté* ("pâté").

Punctuation. Some product names have dots that act as both segmentation characters and truncation signs. In the example 8XDOS.SENSEO CAPP., the words DOS (for *dosette*, "capsule") and SENSEO (brand) should be separated while integrating the period with the word preceding it, contrary to the space, to have the possibility to force the search for the long form of the word thereafter. Indeed, in this example, the lowest distance finds the word *dos* ("back") instead of the word *dosette*, which is impossible because of the presence of the point which means that the word has been abbreviated. However, segmentation should not be applied to the period between two digits, as in the example CRISTALINE 1.5L because it is a decimal number.

Pseudo-syntactic elements. Beyond the word scale reductions and problematic typographic elements previously explained, we also need to analyze reductions at phrase level, which we can consider as multi-word expressions, to allow us to better understand the difficulties of alignment between short forms and long forms. These elements concern the reduction processes used, not from a lexical point of view but from a syntactic and semantic point of view.

Ellipsis. Product names usually include only names and adjectives necessary for product distinction. Prepositions and determinants are omitted, except in rare cases. These ellipses do not prevent the understanding of product names, because these words are almost meaningless. However, some ellipses concern words that are more important for automatic understanding because they carry meaning, like in Example 2: it's semantic ellipses.

Example 2. 205G DESSERT NOIR : *Chocolat noir Nestlé Dessert* ("Dark Chocolate Nestlé Dessert")

Random word order. The order of words in product names is not always intuitive (300G MOUSSAKA BARQ instead of *Barq[uette de] Moussaka*, "tray of Moussaka"). Similarly, the order of words in similar products is not always the same.

Ambiguity of reductions. The reductions can be ambiguous, as we have seen in the state of the art. Thus, we find in our corpus some identical reductions that do not correspond to the same long form. For example, we find PDT for initials of *Pomme de terre* ("Potato") and for syncope of *Président* ("President").

Several reductions for a single long form. For some recurring words, we have noticed the diversity of reduced forms in receipts. For instance, we can find SLD., SAL, SALAD, SLDE and SALADE for *salade* ("salad"). We can notice that some shortened forms are finally not so much reduced in terms of characters. This is the case for example of PIZZ. which contains as many characters as PIZZA.

Spelling variants. Receipts also contain graphic or spelling variants of certain words. Sometimes these variants lengthen the word. For some brands, it is easier to simplify the spelling. This is the case of MINIBABY (for "Mini Babybel"), which is both a contraction of two words and an abbreviation.

3.3 Corpus of product names (possible expansions)

In order to find a less ambiguous form for each reduced forms present on the receipts, we have created a reference corpus containing a list of products whose names are detailed and unabbreviated.

We have thus extracted from the Web a list of 13 888 titles of products distributed by the Drive-in service of a supermarket of the same brand as that of our receipts. The Drive-in service allows you to shop online by choosing products with the information the site provides: detailed product name, price, price per kilogram, product weight, promotions and photography.

The choice of this supermarket brings an advantage and a disadvantage: it allows to have a larger choice of products, and therefore a broader reference corpus, but at the same time, the proposed products are not always the same as those bought in convenience stores. In fact, consumers are not the same in a small local area, located in a student district, where our receipts come from, and in a large area on the outskirts. The products sold are not the same either: we will mainly find in the minimarket meals (salad, sandwiches, ready meals), cupcakes and drinks individually, while we will find more fresh or unprepared products, in family quantity, in the supermarket.

These product names differ a lot from the product names extracted from the receipts. While brevity is required on the receipt, it is detail and accuracy that are most important on the purchase website, to be as exhaustive as possible so that the customer knows which product he buys without having it in front of him. The names of products from the Web are consequently longer (37 characters on average) and contain an average of 5.5 words. Each name product contains a key word, often one or several qualifying terms (adjective or noun) and end with the proper noun of the product or the brand, such as the following example.

Example 3. Céréales goût caramel/chocolat Lion (for "Caramel and chocolate flavored cereals named Lion")

We can however observe some irregularities in these formulas such as repetition of a part of the phrase. There is also some abbreviated words (current initialisms or truncations such as "choco" or "PDT"), ellipses of prepositions ("chocolat lait" instead of "*chocolat au lait*" for "milk chocolate"), numbers and symbols.

4 Proposed approach

We want to find the best long form for each shortened term of our receipts corpus. Matching a shortened term with its full form without context is a very difficult task, especially when the shortened term is so noisy.

4.1 Automatic pre-processing

Our previous analysis allows us to note the main differences between our two datasets. First, the shortened form is totally uppercase and does not contain any diacritics. In order to make the source (receipts terms) and the target (web extracted terms) comparable, we normalized all target characters to capitalize them and delete all diacritics.

Then, we have to segment both receipts terms and web terms into words, considering the difficulties explained below concerning symbols and absence of space. According to the case, we consider one or two tokens around a symbol like "+", "/", ".". We also separate in two tokens the numbers followed by more than 2 letters, such as 4FROMAGES ("*4 fromages*" for "4 cheeses"). We choose the 2-letters limit to avoid to separate the numbers followed by symbol of quantity, like in 75CL ("*75 centilitres*" for "75 centiliters")

We also define the list of the most frequent trigrams of the web terms, to create a dictionary of most probable initialisms. Indeed, we previously noticed that all initialisms of our receipts corpus are composed of three letters, and correspond to frequent multi-word expression, such as PDT. Thus, for each 3-letters word in receipt expression, which is not a real word, we try to figure whether it could be an initialism.

4.2 Automatic matching

Terms of receipts are abbreviated both at the word- and at the phrase-level. This requires to implement a two-step algorithm.

First, we calculate the distance between words of shortened form and each word beginning with the same letter as the product names extracted from the web. We save the words within a short enough distance with the source word, or the initialism. To calculate this distance, we use a weighted Levenshtein distance for the first step, which allows us to put a different weight to each editing operation: substitutions, deletions and insertions. We multiply the cost of each operation by the corresponding chosen weight instead of incrementing by 1. In the case of the search of minimal distance between a shortened form and a long form, we want to have a lower weight for insertion than for the two others. We find after computation that the better weights for deletions and substitutions are 11, leaving insertions with a cost of 1, to obtain the better matching. If no distance is satisfactory, no word is recorded.

For the second step, we propose and compare two different methods. The first one is the weighted Levenshtein distance. From the first step, we obtain a list of probable words that we concatenate to have a phrase we can compare with each web phrase, calculating the distance between them. As for the word level step, we save the expression that have the minimal distance with the source phrase and the best weight is 11. The second one is the Jaccard index, using the intermediary list of words obtained from the first step, as a set. We save the expression that have the higher index (included between 0 and 1).

5 Experiments

5.1 Ground truth

To evaluate our approach, we create a ground truth, manually associating receipts forms and web forms in a JSON format. We encounter some difficulties, because human does not always understand the reduction form, or cannot disambiguate between two similar products with different brands. For example, the PET 1.25L EAU GAZ form should correspond to one item of the 6 full forms list, each corresponding to a different brand, but we can not know which one, and our approach could not either.

For these cases of impossible disambiguation, we propose all the possibilities in our ground truth, to not have false negatives. In some cases, the products on the original receipts are not sold by the Drive-in service, and thus have no shortened form in our corpus. Such cases are evidently discarded in the evaluation.

5.2 Results and discussions

Results are measured based on the binary match between the proposed long expression and the ground truth, which can return “True” or “False”, depending on whether the proposed long expression is the correct one or not. In addition, cases when no ground truth is associated to the abbreviation are ignored (cases that were not annotated or cases when even the human evaluator cannot perform the task). Thus we compute a percentage of correct answers, which is 36,87% for the best combination of weights in weighted Levenshtein method, and 33,64% for the Jaccard index method.

We observe that the Jaccard approach does not perform as well as the method based on weighted Levenshtein. This is due to marks of plural on many words: while the edit distance increases by 1, the Jaccard index does not recognize the same word. For example, OIGNON JAUNE, corresponding to *Oignons jaunes* (plural) in web source (for “Yellow onion”), with the intermediary list of words containing [OIGNON (singular), JAUNE (singular)]. The Jaccard index between the intermediary word set and the web word set is zero, resulting in a wrong answer. The weighted Levenshtein gives a score of 2 and provides the correct answer.

The traditional measures, namely Precision, Recall and F-Measure, are not the most appropriate to evaluate the performance of our system, because there is no prediction, only scores corresponding to distances. If we select the criteria $distance = 1$ for correct prediction and $distance \neq 1$ for wrong prediction, we obtain the results presented in Table 4. We only see with the first two results that when the distance is small, the result of the algorithm is always correct, but there are very few such cases. The next two lines show the maximal F-Measure we can find by changing the criteria of selection of relevant elements.

Table 4. Evaluation with traditional measures

Method	Precision	Recall	F-measure	Rejection	Accuracy
Jaccard index = 1	1.0	0.06	0.10	1.0	0.68
Weighted-Levenshtein distance = 1	1.0	0.04	0.07	1.0	0.65
Jaccard index $\geq$ 0.33	0.60	0.79	0.68	0.74	0.76
Weighted-Levenshtein distance $\leq$ 50	0.47	0.69	0.56	0.54	0.59

Given that these evaluation results are not very conclusive, we propose an evaluation approach that allow a more qualitative evaluation of results. Only one third of our shortened phrase corpus is correctly found, but, when we analyze the results, we see that performance is not as weak, since the failed word is often the brand name, and most of the phrase if often correctly identified.

The Table 5 shows the average of the scores of three distances computed between the proposed phrases and the ground truth according to the method used. We can observe that there are only 16 edit operations between the two strings and that almost half of the words of the whole two phrases are common. The third line of Table 5 concerns the following test : we give the receipts forms to a French human and he has to find by himself what it could mean and to write the name of the product. We observe that the distance is more important than with our algorithm. This is explained by the addition of many keywords describing the product, as "*boîte de conserve*" (for "can").

Table 5. Evaluation with distance measures

Method	Success rate	Jaccard	Overlap	Levenshtein
Jaccard	33.64	0.44	0.45	15.98
Weighted-Levenshtein	36.87	0.49	0.43	15.46
Human		0.19	0.06	18.34

6 Conclusion

The heterogeneity of product name reductions, whether in the diversity of reduced forms or more generally in the variety of reduction processes, the presence or absence of punctuation or the mixture of special characters, numbers and letters, makes rich the corpus analysis and complex its automatic processing. The automatic matching of shortened words and phrases with their expansion without context is an undeniable challenge. Indeed, the biggest part of people searching for matching abbreviations and expansions uses context to find relations between them in natural language or to disambiguate the meaning.

Qualitative analysis of our experimental results are very promising and we think that the combination of the presented approach with an automatic analysis of contextual elements (like other products bought, hours of shopping...) would allow us to improve the choice of the better expansion for each shortened form.

To foster further research, we make the dataset and its ground truth available to other researchers within an ICPR 2018 competition[5].

References

1. Riegel, M., Pellat, J.C., Rioul, R.: Grammaire methodique du francais. Presses universitaires de France (2016)
2. Grevisse, M., Lits, M.: Le petit Grevisse: Grammaire française. Grevisse Langue Française. De Boeck Secondaire (2009)
3. Martinet, A.: Éléments de Linguistique Générale ... Collection Armand Colin, no. 340. Section de litérature. Librairie Armand Colin (1967)
4. Yeates, S.: Automatic extraction of acronyms from text. In: University of Waikato. (1999) 117–124
5. Yu, M., Li, G., Deng, D., Feng, J.: String similarity search and join: a survey. **10**(3) (2016) 399–417
6. Larkey, L.S., Ogilvie, P., Price, M.A., Tamilio, B.: Acrophile: An automated acronym extractor and server. In: Proceedings of the ACM Fifth International Conference on Digital Libraries, DL 00, Dallas TX, ACM Press (2000) 205–214
7. Nadeau, D., Turney, P.D.: A supervised learning approach to acronym identification. In: Proceedings of the 18th Canadian Society Conference on Advances in Artificial Intelligence. AI'05, Berlin, Heidelberg, Springer-Verlag (2005) 319–329
8. Gaudan, S., Kirsch, H., Rebholz-Schuhmann, D.: Resolving abbreviations to their senses in medline. Bioinformatics **21**(18) (2005) 3658–3664
9. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. Volume 10. (1966) 707–710

[5] Fraud Detection Contest: Find it!: `http://findit.univ-lr.fr`