ReSyf: a French lexicon with ranked synonyms

Billami, Mokhtar B.
Aix-Marseille University
LIS UMR 7020
Marseille, France
mokhtar.billami@univ-amu.fr

François, Thomas
Catholic University of Louvain
FNRS-CENTAL/IL&C
Louvain-la-Neuve, Belgium
thomas.francois@uclouvain.be

Gala, Núria
Aix-Marseille University
LPL UMR 7309
Aix-en-Provence, France
nuria.gala@univ-amu.fr

Abstract

In this article, we present ReSyf, a lexical resource of monolingual synonyms ranked according to their difficulty to be read and understood by native learners of French. The synonyms come from an existing lexical network and they have been semantically disambiguated and refined. A ranking algorithm, based on a wide range of linguistic features and validated through an evaluation campaign with human annotators, automatically sorts the synonyms corresponding to a given word sense by reading difficulty. ReSyf is freely available and will be integrated into a web platform for reading assistance. It can also be applied to perform lexical simplification of French texts.

1 Introduction

With the availability of very large corpora and the growing maturity of corpus linguistics and NLP techniques, quantitative descriptions of the lexicon have made noteworthy progress. The coverage of the resources has increased, tokenizing and part-of-speech tagging have enabled a better annotation of the lexical units and statistical models have yielded more accurate descriptions of the lexicon (in terms of frequencies, n-gram models, etc.). However, to the best of our knowledge, on-line lexical resources with a ranking of the lexical units as regards to their difficulty to be read and understood are scarse. Such information is highly relevant in communicative and educational contexts (e.g. clear and efficient writing, calibration of teaching materials, etc.) given that complex synonyms are identified and their simpler equivalents proposed.

In this paper, we present and describe such a resource for French. In ReSyf, synonyms are automatically disambiguated and ranked according to their complexity for L1 schoolchildren readers. The lexicon has been developed as part of the project ALECTOR[1] (Reading Aids to leverage Document Accessibility for Children with Dyslexia) that aims to support poor readers and dyslexic children to acquire French vocabulary and improve their reading skills in French through practise. More specifically, ALECTOR addresses the challenge of automatic text simplification for this population and illustrates one of the possible uses of ReSyf: it can be a knowledge database to carry out the substitution of complex words present in reading materials. The evaluation of the proposed ranking algorithm of the synonyms shows that, in 91% of the cases, the ranks in ReSyf correspond to the ranks given by human annotators. These results are encouraging taking into account the difficulty of the task (i.e. subjectivity, inter-annotator agreement). They are also important because the simplest synonym is always identified, which implies that ReSyf can indeed be used for reading assistance and can also be integrated into a model for automatic lexical simplification.

The paper is organized as follows. After reviewing related work at Section 2, we present the methodology applied to create ReSyf at Section 3. It includes three steps: (1) the choice of the synonym resource; (2) the disambiguation process, and (3) the ranking algorithm. Section 4 provides details on the resource itself, its availability and its evaluation by humans. Some concluding remarks and future work are discussed in Section 5.

[1] https://alectorsite.wordpress.com

Proceedings of the 27th International Conference on Computational Linguistics, pages 2570–2581
Santa Fe, New Mexico, USA, August 20-26, 2018.

2 Related work

Despite playing a crucial role in semantic applications, available machine-readable resources with synonyms are still scarce. In English, the Roget's 21st Century Thesaurus[2] (third edition) and WordNet (Miller et al., 1990) are notable examples. In French, the lexicon from CRISCO[3] (Morel and François, 2015) and the lexical network JeuxDeMots[4] (Lafourcade, 2007) are the most well-known and freely available on-line resources. Other semantic resources including synonymy relations are WOLF – the French WordNet – (Sagot and Fiser, 2008) or BabelNet (Navigli and Ponzetto, 2012). Yet, they have been automatically built from multilingual networks and therefore the granularity of their senses often lacks precision to be exploited for our purposes (not to mention wrong translations in the case of the French version of BabelNet).

With regards to graded lexicons, the idea of assessing a degree of 'difficulty' to words – although not recent – is not widespread. In the context of language learning and reading assistance (i.e. automatic lexical simplification in NLP), 'difficulty' can be understood as the level of proficieny required for a learner to read and understand a word. Beyond frequency counts, commonly used as proxys of the familiarity of a word[5], some initiatives have recently taken advantage of corpora with levels of difficulty. For French, Manulex (Lété et al., 2004) is a graded lexicon used by teachers and psycholinguists to identify the level of difficulty of words at school (in the context of learning French as mother tongue). This lexicon describes the frequency distributions of 23,812 French lemmas across three primary school levels: 1st grade, 2nd grade, and 3rd to 5th grades. The frequencies have been estimated on a corpus of pedagogical materials used at these three levels. FLELex (François et al., 2014) is a similar resource, aimed at learners of French as a foreign language. It is part of a larger project including graded resources for different European languages: the CEFRLex project[6]. All the resources in the CEFRLex project have been built based on corpora of pedagogical texts classified according to the six proficiency levels defined by the CECR (Conseil de l'Europe, 2001), ranging from A1 to C2.

The difficulty information in Manulex and FLELex can directly be used to assign difficulty levels to French words and synonyms (Gala et al., 2014). However, this approach presents some drawbacks. First, as they are both pedagogical resources, their coverage is limited. We thus have no level information for more complex synonyms if they are absent from these resources. Second, the pedagogical levels used in these resources may sometimes be too coarse-grained to make fine-grained distinctions as regards the complexity of synonyms. For instance, the word *minceur* ('thinness, slimness') is assigned the difficulty level 3 in Manulex, which is the highest difficulty level in this resource. However, in ReSyf, *minceur* is ranked at the 6th position in a list of 12 synonyms. Synonyms from position 7 to 12 are assumed to be more complex than *minceur*, but if we assign them the difficulty scores from Manulex, they would all be assigned to level 3. Such coarse-grained information does not allow to make any finer difficulty distinction between all these synonyms. We therefore believe that ranking synonyms is more efficient than assigning coarse-grained difficulty levels. Third, synonyms are extracted – and their difficulty is estimated – as word forms and not as word senses. For instance, *mince* is an adjective meaning 'thin, faint, slim'; it is given grade 1 in Manulex and A1 in FLELex, while in ReSyf it appears in 1st position with the meaning 'faint', in 3rd position with the meaning 'slim', and in 4th position with the meaning 'thin'.

As far as specific resources for automatic lexical simplification are concerned, to the best of our knowledge only the lexicon CASSAurus (Baeza-Yates et al., 2015) for Spanish is similar to ReSyf. However, the words in this lexicon are assigned only to two classes, simple and complex, which is a very coarse-grained view of lexical complexity.

[2] http://www.thesaurus.com

[3] http://www.crisco.unicaen.fr/des/

[4] http://www.jeuxdemots.org

[5] The first frequency lists appeared nearly one hundred years ago, among which: Thorndike (1921), Ogden (1930), Tharp (1939), Gougenheim (1958).

[6] http://cental.uclouvain.be/cefrlex/

3 Methodology to build the lexicon

To build ReSyf, we used the words in the lexical network JeuxDeMots (Lafourcade, 2007) linked by the synonymy relation (3.1). We then applied a disambiguation (3.2) and a ranking algorithm (3.3) to identify complexity within the word senses in a vector.

3.1 Nature of the lexical entries: synonyms

Synonymy is a lexico-semantic relation implying equivalence among two word senses. Absolute synonyms are very rare, synonyms are thus two lexical units having a semantic value close enough (by inclusion or intersection) to be replaced one by the other to convey the same meaning (Polguère, 2002). Word Sense Disambiguation (WSD) techniques are necessary to identify the meaning of the word forms to be able to identify the appropriate synonyms, i.e. the French word *grave* is synonym of *sérieux* when meaning 'serious' (i.e. a serious problem), but synonym of *profond* when meaning 'deep' (i.e. a deep voice).

3.2 Data acquisition: from word forms to word senses

3.2.1 Resources for bootstrapping

Similar to other resources where the complexity of the entries is given (see section 2), we first developped a version of ReSyf with word forms associated with grades (from 1 to 3) (Gala et al., 2013) and (Gala et al., 2015). This approach raised some important issues concerning the nature of the entries and Word Sense Disambiguation. These problems were very soon identified and we started to work in another version with refined senses[7] for each entry and with ranks dynamically assigned from 1 to n, depending of the number of synonyms in a vector, instead of static grades.

We created a second version using BabelNet (Navigli and Ponzetto, 2012), a lexical network automatically built from several resources (WordNet, Wikipedia, etc.). The problem with BabelNet is its excessive granularity of senses: for our purposes (for the targeted users of ReSyf), we needed a more accurate and simplified version. For instance, the word *toile* in French has fifteen senses in BabelNet. For schoolchildren, specially those with reading difficulties, four distinctions are enough ('computer network' , 'cloth' , 'painting' , 'spider web').

We finally decided to use the lexical network JeuxDeMots (Lafourcade, 2007) for the current version. The lexical database of this network, built by crowdsourcing, is very rich and constantly evolving. Each node in the lexical network represents a lexical unit describing a word (a simple word or a multiword expression, MWE). The relationships between the nodes are typed and weighted. Some of these relationships correspond to lexical functions related to the vocabulary itself (such as ASSOCIATED IDEA and SYNONYMY relationship) or to hierarchical semantic relations (such as HYPERNYMY and HYPONYMY). Senses are encoded with the SEMANTIC REFINEMENT relation. Each instance of this relation describes a specific sense for a given term.

JeuxDeMots is a human-based computation game, in other words, a game with a purpose (GWAP). The actors are simple players who play through an interface that has the form of an online game. The validation of the quality of the data collected for the construction of the lexical database, containing lexical items and relations between them, is provided by players. More specifically, relationships are proposed anonymously by a player and they are validated by other anonymous players. The relations between the lexical items are weighted. The weighting is carried out in the following way: the more an instance of a given relation is proposed, the more its weight increases, as long as the conditions of the game are respected.

At present, the JeuxDeMots lexical database contains:

a) 182 373 582 instances of all possible relations;

b) 3 081 091 terms having at least one outgoing relation ($term_A \rightarrow term_B$);

c) 2 497 238 terms having at least one incoming relation ($term_A \leftarrow term_B$).

[7] A refined sense $s_i(w)$ for the word w is a particular use of w that this word has no other similar refined sense to $s_i(w)$.

Synonyms are proposed for both words and word senses. However, it turns out that not all synonyms for a given word are grouped together in distinct senses. For example, *tissu* ('cloth') is a synonym for the lexical entry *toile* but it does not appear in the network as a disambiguated synonym. We thus need to include *tissu* as a disambiguated synonym aggregated to the word sense *toile* ('cloth'). To include all disambiguated synonyms into word senses, we use a disambiguation algorithm based on semantic representations for words and word senses (semantic signatures).

We previously proposed a methodology for the creation and validation of semantic signatures (Billami and Gala, 2017). In general terms, a semantic signature can be considered as a special representation form of Vector Space Model, VSM (Turney and Pantel, 2010). In the same way as VSM, the weight associated with a dimension in a semantic signature indicates relevance or importance of this dimension. The main difference is the way in which the weights are computed. VSM uses cooccurrence statistics in a given corpus whereas semantic signature uses structural properties of the used network (JeuxDeMots in our case).

The goal of our approach is to directly compare the semantic signature of each synonym with the semantic signature of each candidate word sense. The semantic similarity that we use is an activation function which takes into account the relation between two lexical units to compare (Billami and Gala, 2017). This relation checks whether the one (synonym or candidate word sense) represent a dimension in the semantic signature for the other. If it is true, the function returns a perfect similarity ($score\ of\ similarity\ =\ 1$) else the cosine similarity is estimated by using their semantic signatures.

We use the relation ASSOCIATED IDEA for creating the signatures. This relation contains the largest number of instances since it includes all terms that reflect a given lexical entry in JeuxDeMots (57% for ASSOCIATED IDEA instances). On the other hand, JeuxDeMots contains at least 100 lexico-semantic relations.

3.2.2 Word Sense Disambiguation method

The algorithm 1 described below allows to disambiguate a synonym by choosing the most suitable target word sense. For each pair of synonym syn_a and candidate word sense $Sense_i$ with $i \in 1 \ldots$ n (n : number of senses for the target word) the algorithm first checks whether the one represent a dimension in the semantic signature for the other. If it is true, the synonym is clustered directly to the $Sense_i$ else the cosine similarity is estimated by using their semantic signatures. In this case, the closest sense(s) with the best similarity are selected. We use a threshold $\varepsilon = 0.01$.

We have developed a variant of the algorithm 1 by comparing not directly a synonym with a candidate word sense, but each synonym sense (if this latter is polysemous) with each candidate word sense. This is the hypothesis that describes the similarity between two words as the similarity of their closest senses (Budanitsky and Hirst, 2006). On the other hand, if the synonym is monosemous, the algorithm 1 is applied (we note this variant algorithm 2).

In order to validate our algorithms, we used a list of manual disambiguated synonyms in JeuxDeMots as a test set (JeuxDeMots dump of December 2017, 33 039 pairs synonym – target word sense). The table 1 describes the precision results of our evaluation by applying the two clustering algorithms. In Word Sense Disambiguation, the precision is the ratio between the correct answers provided and the total answers provided, whereas recall is the ratio between the correct answers provided and the total answers to provide (Navigli, 2009).

Our semantic signatures based on associated ideas cover all synonyms and senses of the test set. Therefore, we have precision = recall = F-measure.

Not surprisingly, the results show that we have a better performance (higher precision) when we use the algorithm 1: the semantic signature of a synonym is most informative than a semantic signature for a specific synonym sense.

3.3 Data ranking method: from grades to ranks

Once all the senses were identified, we developed a ranking algorithm which is able to sort the synonyms according to their complexity (reading difficulty for target readers). For this task, we relied on an approach commonly used in the field of information retrieval to sort the results of a query by

```
Algorithm 1: Comparison of each candidate word sense with each synonym $syn_a$
Input:
$target\ word$: word to treat
$sem\_ref\ (target\ word)$: set of senses of the target word
$syn_a$: synonym of the target word
$\varepsilon$: validation threshold of the similarity
Result:
$\mathbf{\hat{Sense}_{target\ word}}$: senses of the target word with the highest score
Data:
$\mathrm{S_{a\_idea}}$: set of signatures whose dimensions are associated ideas
Initialization:
$Score_{refs\_C} = \emptyset$ // Score of the target word senses
for $Sense_i \in sem\_ref\ (target\ word)$ do
    $Score = \begin{cases} 1\ if\ (*) \\ Cosine\ (S_{a\_idea}(Sense_i), S_{a\_idea}(syn_a))\ otherwise \end{cases}$
    $(*) : Sense_i \in S_{a\_idea}(syn_a) \vee syn_a \in S_{a\_idea}(Sense_i)$;
    if $(Score \geqslant \varepsilon)$ then
        $Score_{refs\_C} \leftarrow Score_{refs\_C} \bigcup (Sense_i,\ Score)$;
$\mathbf{\hat{Sense}_{target\ word}} \leftarrow$ Best $(Score_{refs\_C})$
```

Clustering algorithms	Correct annotations	All annotations	%
Algorithm 1	32 802	33 039	**99.28**
Algorithm 2	25 307	33 039	76.6

Table 1: Evaluation results by applying the two clustering algorithms.

relevance: the pairwise approach, and more specifically the SVMRank algorithm (Herbrich et al., 2000). Such an approach requires a database of words already sorted by difficulty that will be used to create a training dataset composed of pairs of words with different levels of difficulty (section 3.3.1). We also computed, for each pair, a set of linguistic features (section 3.3.2) that can be used to predict which word of the pair is the most complex one. After model optimisation (section 3.3.3), we obtained a function that can predict, for a given input (word sense pair), which one is the more complex. This function may then be integrated into any sorting algorithm to rank a set of synonyms.

3.3.1 Training dataset

As ReSyf is mainly intended for schoolchildren, we used Manulex to obtain word difficulty annotations (see description at Section 2). As we only considered open class words (nouns, adjectives, adverbs, and verbs), we retained $19,038$ lemmas from Manulex for our training dataset.

Based on this resource, it is possible to create pairs of words in which one is more complex than the other. The pairs are then used to train the ranking model. However, this requires to transform the frequence distribution of each word into a single numerical value that can be used to compare the reading difficulty of two words. More formally, the distribution D for a word w takes the form of a vector (f_1, f_2, f_3) corresponding to the frequencies of the word at each of the three Manulex levels. Our goal is to define a function $\phi(D)$ that will output a single difficulty value l based on the values in D. Two approaches were tested. The first technique simply outputs a level value $L \in \{1, 2, 3\}$ that corresponds to the first level f_i for which $f_i > 0$. The training set based on this technique is called *Manulex-3N*. However, using only three values to represent difficulty creates a large amount of ties during the pair creation step. Therefore, we experimented a second technique to define $\phi(D)$ such as it outputs a continuous value ranging from 1 to 3 using the formula below (Gala et al., 2013).

$$\phi(D) = L + \exp^{-r} \quad \text{where} \quad r = \frac{\sum_{i=1}^{L} f_i}{\sum_{i=L+1}^{3} f_i}$$

In this formula, L corresponds to the output of the first technique to which we add a continuous quantity that is function of the distribution D. This quantity is defined in terms of the ratio between the sum of the counts from levels 1 to L over the sum of the counts from $L + 1$ to 3. This way, we can distinguish two words such as *pomme* ('apple') et *cambrioleur* ('burglar') that both appear at level 1 ($L = 1$), but 724 times for 'apple' and only 2 times for 'burglar'. The training set based on this technique is called *Manulex-Cont*.

3.3.2 Word features

Each word sense from our dataset was first represented as a 69-feature vector capturing various linguistic and psycholinguistic properties that can be classified in the four following types:

- *Spelling features*: (1) number of letters, (2) number of phonemes, (3) number of syllables, (4) number of orthographical neighbors[8]; (5) cumulated frequencies of all orthographical neighbors; (6) number of neighbors that are more frequent as the target word; (7) transparency between the written and phonological forms; (8-13) six variables detecting specific complex graphems, namely oral vowels (e.g. *au* [o]), nasal vowels (e.g. *in* [ɛ̃]) , double consonnants (e.g. *pp*), double vowels (e.g. *ée*), other digrams (e.g. *ch* [ʃ]), or the sum of all five phenomenas; and (14-16) membership of the syllabic structure of the word to a class considered as either frequent, median, or rare.

- *Frequency features*: (17) the log-frequency of the word based on the Lexique3 (New et al., 2007) database and (18-26) the presence of the word in a list of simple words. We defined 9 lists of different sizes (1063, 2000, 3000, $\cdots$, 8000, and 8775 words), all based on the Gougenheim list (Gougenheim, 1958).

- *Semantic features*: (27) a binary variable coding whether the word is considered as polysemic in JeuxDeMots and (28) the number of synsets listed in BabelNet.

- *Morphological features*: the morphological analysis was automatically performed by systems developed by Bernhard (2006) and Bernhard (2010). The first of these systems splits words into tagged morphemes (root, prefixes, suffixes, etc.) and derives various frequency information about the identified morphemes, while the second system identifies the morphological families and can be used to extract information about a word family. The variables we used were : number of morphemes; presence of suffixes; presence of prefixes; presence of two bases or more (for compound words); the minimal frequency of the affixes of the word (i.e. number of different words in which appears the least frequent of the affix); the average frequency of all affixes in the word; the size of the morphological family; the frequency of the most frequent word in the morphological family; the mean frequency of all words in the family; and the cumulated frequency of all words in the family. The two systems include parameters that were manipulated, thus creating variants of the above variables. In total, we defined 41 morphological variables (29-69)[9].

3.3.3 Model definition and optimization

The definition of the ranking model was performed in three steps: (1) creating the training dataset (pairs of synonyms); (2) feature selecting; and (3) model training. To create the pair training dataset, we applied the following procedure: given two words w_i and w_j, each associated to a difficulty level (l_i and l_j) and to a feature vector ($\mathbf{v_i}$ and $\mathbf{v_j}$), we create a pair $< w_i, w_j >$ for which a new vector $\mathbf{v_{ij}}$ is obtained from the combination of the two vectors $\mathbf{v_i}$ and $\mathbf{v_j}$. Several arithmetic operations can be used to carry out

[8] The orthographic neighborhood of a word have been defined by Coltheart (1978) as all the words of same length and differing only by one letter (eg. FIST and GIST).

[9] For a detailed description of these parameters, see (Gala et al., 2014).

this combination, but we used substraction ($\mathbf{v_{ij}} = \mathbf{v_i} - \mathbf{v_j}$), as Tanaka-Ishii et al. (2010) showed that substraction was best for ranking texts by readability. Each pair $< w_i, w_j >$ was also assigned a new difficulty level (l_{ij}) obtained with the following rule: (1) if $l_i > l_j$, then $l_{ij} = 1$ and (2) if $l_i < l_j$, then $l_{ij} = -1$. As we had two original datasets, *Manulex-3N* and *Manulex-Cont*, we got two pair datasets.

Second, to select the best predictors of word difficulty, we computed the Spearman correlations between each of our 69 variables and the new binary difficulty variable (L_{ij}). We used only the *Manulex-3N* dataset for this aim. Table 2 displays the correlation for some of the best features in our set. At the end of this selection step, 21 variables were retained for the model.

Variable name	**Correlation (ρ)**
17 Freq. Lex3	– 0.57
18 AbsGoug (6000)	– 0.46
02 Nb. phon	0.35
15 Polysemy	– 0.33
01 Nb. letters	0.32
03 Nb. syllables	0.32
4a Nb. neighbors	– 0.23
15 Mean freq. of the morphological family	– 0.27
15 Cum. freq. of the morphological family	– 0.27
15 Max. freq. of the morphological family	– 0.27
4b Cum. freq. of the neighbors	– 0.23
16 Nb. of senses in BabelNet	– 0.19

Table 2: Best variables based on Spearman correlation.

Third, we trained a SVM model with linear kernel on each pair dataset (*Manulex-3N* and *Manulex-Cont*). For each model, a grid-search was used to select the best value for the C meta-parameter. We then estimated the accuracy of pair classification by each model with a 10-fold cross-validation procedure. Table 3 summarizes the accuracy of both models and compares their scores with those of a full model (that uses all 69 features). The models including 21 variables reach similar or slighly better accuracy than the full model. More interestingly, the model trained on the *Manulex-3N* dataset (using the simple rule of first occurrence) clearly outperforms the model based on *Manulex-Cont*. As a result, we decided to retain the model based on the first way of defining $\phi(D)$ for the final version of ReSyf.

Dataset	**C**	**21 var.**	**C**	**69 var.**
Manulex-3N	0.01	77.4%	0.01	77.8%
Manulex-Cont	0.01	72.4%	0.01	71.4%

Table 3: Accuracy of the ranking models.

4 ReSyf: graded synonyms according to their difficulty

In this section we describe the data available in the lexicon (section 4.1) and an evaluation of the ranking algorithm (section 4.2).

4.1 Data available

ReSyf provides an inventory (a vector) of equivalent words ranked according to their difficulty to be read and understood[10]. For instance, *sec*(1), *léger*(2), *mince*(3), *allongé*(4) and *svelte*(5), corresponding to the meaning 'slim' . The first sense *mince (fin)* is the most general sense taking into account the weight of the relation between the word *mince* and the semantic refinement *mince (fin)* defined in JeuxDeMots. The weight of this type of relation allows to sort the senses from the most general to the most specific. Each weight is normalized by using the ratio between its value and the top level weight value.

[10] The resource is freely available for lookup and download (XML file) at http://cental.uclouvain.be/resyf

In order to distinguish words according to the four open classes (nouns, adjectives, adverbs, and verbs), we filtered all the single words with the French reference resource Lexique3 (New et al., 2007). For the multiword expressions (MWE), we used the parser Talismane[11] (Urieli, 2013) as a part-of-speech tagger. For MWE, we have considered that its POS is the POS assigned to its first open class item. Table 6 describes the distribution of entries in ReSyf (total number of entries: 57 589, 10 333 polysemic and 47 256 monosemic). The number of common nouns is greater than that of all the other parts-of-speech categories, either for the polysemic or the monosemic entries. Note that JeuxDeMots is constantly evolving and, as a result, more semantic refinements will be available in the future (more polysemic entries to be disambiguated).

The distribution of categories described in table 6 shows that the mean synonyms per polysemic entry sense is 4.95 (which is greater than what we can obtain when using other resources such as BabelNet).

	Nouns	**Verbs**	**Adjectives**	**Adverbs**	**Total**
#Polysemic entry (*Pe*)	**6 737**	1 779	1 691	126	**10 333**
#Monosemic entry (*Me*)	**30 869**	8 388	6 606	1 393	**47 256**
Single words	**21 495**	5 065	7 635	1 105	**35 300**
Multiword expressions	**16 111**	5 102	662	414	**22 289**
Mean synonyms per *Pe*	12.95	**17.97**	16.93	6.16	14.39
Mean synonyms per *Me*	4.19	6.86	**9.27**	4.71	5.39
Mean senses per *Pe*	2.95	**3.03**	2.65	2.25	2.9
Mean synonyms per *Pe* sense	4.39	5.92	**6.39**	2.73	4.95

Table 4: Distribution of ReSyf entries.

Table 5 describes the statistics as regards to the distribution of ReSyf synonym annotations (polysemic entries). As it is showed, JeuxDeMots currently proposes 27 466 associated synonyms to the semantic refinements for adjectives, adverbs, common nouns and verbs. By applying our first clustering algorithm (cf. algorithm 1), we are able to automatically disambiguate 121 182 synonyms. The automatic annotation represent a benefit of 4.4 more than what JeuxDeMots currently proposes.

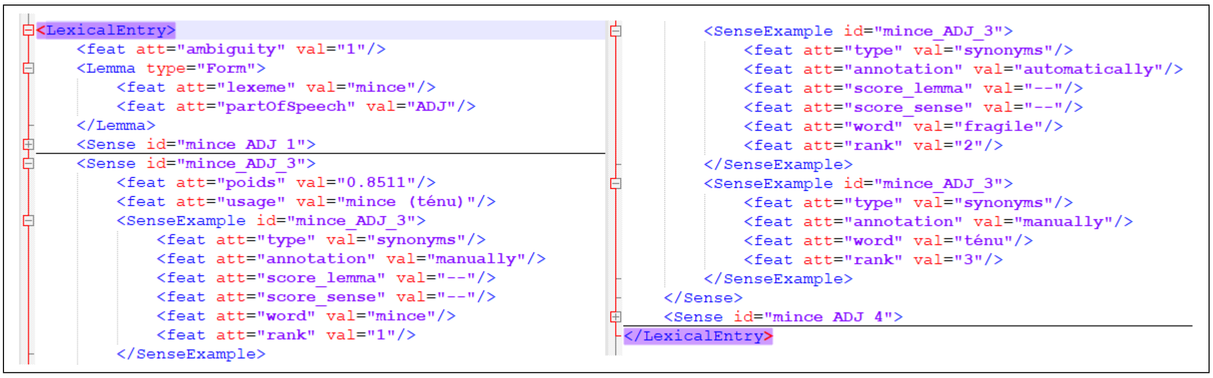
	Nouns	**Verbs**	**Adjectives**	**Adverbs**	**Total**
#Automatic annotations	**69 323**	26 379	24 851	629	**121 182**
#Manual annotations	**17 954**	5 584	3 781	147	**27 466**

Table 5: Distribution of ReSyf synonym annotations (all polysemic entries).

The figure 1 shows a sample description of the Lexical Markup Framework (LMF) format used to encode the data in a XML file. Each lexical entry is encoded in the node 'LexicalEntry'. This latter contains different features such as the ambiguity (which takes 'one' if the lexical entry is polysemic or 'zero' otherwise), the lemma form to encode the lemma and the part of speech of the entry and sense nodes to encode all senses.

The figure 1 shows all the details for the second sense of *mince* ('faint, delicate'). The weight of this sense is 0.8511, which is lower than that of the first sense ('thin', 1.0). Each sense is available with its weight, its usage features and some examples which define the disambiguated synonym. Each sense example contains the annotation feature which takes a 'manually' value if the synonym is defined manually or 'automatically' value if the synonym is captured by the clustering algorithm 1. The feature 'word' encodes the lemma of the synonym. The features 'score_lemma' and 'score_sense' encode the maximal score obtained by our clustering algorithms. If these scores are greater or equal to 0.5 they are printed, else we print only '- -'.

[11] http://redac.univ-tlse2.fr/applications/talismane/talismane_en.html



```
<LexicalEntry>
    <feat att="ambiguity" val="1"/>
    <Lemma type="Form">
        <feat att="lexeme" val="mince"/>
        <feat att="partOfSpeech" val="ADJ"/>
    </Lemma>
    <Sense id="mince ADJ 1">
    <Sense id="mince_ADJ_3">
        <feat att="poids" val="0.8511"/>
        <feat att="usage" val="mince (ténu)"/>
        <SenseExample id="mince_ADJ_3">
            <feat att="type" val="synonyms"/>
            <feat att="annotation" val="manually"/>
            <feat att="score_lemma" val="--"/>
            <feat att="score_sense" val="--"/>
            <feat att="word" val="mince"/>
            <feat att="rank" val="1"/>
        </SenseExample>
        <SenseExample id="mince_ADJ_3">
            <feat att="type" val="synonyms"/>
            <feat att="annotation" val="automatically"/>
            <feat att="score_lemma" val="--"/>
            <feat att="score_sense" val="--"/>
            <feat att="word" val="fragile"/>
            <feat att="rank" val="2"/>
        </SenseExample>
        <SenseExample id="mince_ADJ_3">
            <feat att="type" val="synonyms"/>
            <feat att="annotation" val="manually"/>
            <feat att="word" val="ténu"/>
            <feat att="rank" val="3"/>
        </SenseExample>
    </Sense>
    <Sense id="mince ADJ 4">
</LexicalEntry>
```

Figure 1: Lexical entry description of the word *mince* using the LMF format where the second sense is developed.

4.2 Comparison of automatic ranks with human judgments

We carried out an evaluation campaign with forty human annotators in order to obtain a gold-standard to evaluate our ranking algorithm. The annotators were asked to manually rank a list of forty senses (vectors) containing 2 to 6 synonyms (average of 3.5 synonyms/vector), with a total of 150 word forms (53 % nouns, 23 % verbs and adjectives, 1 % adverbs). The synonyms were proposed randomly (in terms of difficulty) and were not contextualized.

The ranks were manually annotated by 28 native speakers of French and 12 C1/C2 non-natives living in France for more than 5 years and having another Romance language as mother tongue. All of them were adults in the academic field: master or PhD students, assistant professors and researchers, with an average of 28,23 years old (standard deviation of 10.07).

The final reference list[12] obtained after the annotations contains 134 word forms and 36 meanings (4 senses corresponding to 16 synonyms were removed from the original list because (a) equality of the annotations or (b) presence of an unknown or irrelevant term in the vector, judged as so by most than one third of the annotators).

For each vector, the Krippendorff's alpha coefficient (α) was calculated. The global agreement obtained is 0.4, it slightly varies when it is calculated specifically for vectors with 3 or 5 synonyms. Unsurprisingly, the lesser the synonyms to rank, the higher the coefficient. These results can be compared with those obtained by Specia and collaborators at SemEval 2012 (Specia et al., 2012) (κ = 0.386 and 0.398) for a similar task.

The ranking algorithm we have developed achieves encouraging results: 83.33 % of the vectors are sorted exactly as the human annotators did, or with a slightly difference of one rank. Only 16.67 % of the vectors show a couple of synonyms ranked with more than two ranks of difference.

In terms of lexical units (synonyms), 91.04 % of them are correctly sorted or inversed with only one rank, 8.96 % have been automatically ranked with a difference of two ranks and 2.24 % (3 synonyms) have been sorted with a distance higher than two. This precise case is that of the adjectives *merveilleux, fantastique, fabuleux, formidable, splendide* ('marvellous, fantastic, fabulous, wonderful, splendid') were the annotators mostly proposed *fabuleux, formidable, fantastique, splendide, merveilleux*. This example shows the difficulty of the task for word forms with similar formal features (length, number of syllables, presence of digraphs) and corresponding to subjective senses with already very low human agreement (Krippendorff's α = 0.04).

5 Conclusion

In this paper, we have presented ReSyf, a resource for French with disambiguated synonyms that have been sorted according to readability features. The results of the ranking algorithm have been compared to human annotations and in 91% of the cases the synonyms are automatically ranked from the simplest

[12] Avalible at the end of the paper (Appendix A).

to the more complex. The lexicon can be used on-line, yet our aim is also to integrate ReSyf into a lexical simplification algorithm in order to reduce the lexical complexity of texts (lexical substitution task).

The perspectives of our work are twofold. First, we are working on the refinement of the lexicon in order to include multiword expressions (MWE). For now, the ranking of MWEs is based on an average of its content words (eg. for *faire faux bond* meaning 'to let down' , literally 'to make a false jump' , the ranking is based on feature vectors for *faux* and *bond*, which is an approximation of the reality). The issue of MWE handling is crucial for NLP applications, a more precise estimation of their complexity in reading comprehension is a real challenge.

A second perspective is that of tailoring the lexicon to the special needs of particular target audiences. By integrating ReSyf into an automatic text simplification system (lexical substitution), we aim at providing a tool to automatically adapt the words in a text to the specificities of a target reader. Roughly speaking, if avoiding long words is recommended for people with dyslexia, it might be interesting to favor them in texts tailored to adults with vision problems such as age-related macular degeneration (AMD). For instance, replace *cambrioleur* by *voleur* ('burglar' , 'thief') for dyslexic readers, but keep *cambrioleur* for AMD patients. Lexical simplification could find useful applications for different populations who struggle with reading. ReSyf could be a key component for such automatic simplification, but also for teachers, speech therapists and other professionals involved in vocabulary learning and reading assistance.

Acknowledgements

This work has been funded by the French Agence Nationale pour la Recherche, through the ALECTOR project (ANR-16-CE28-0005), and by the Belgian National Agency for Research (FNRS). The current version of the ReSyf website has been funded by Ortolang, we thank Dorian Ricci (UCL) and Brayan Delmée (UCL) for their work. We also thank Firas Hmida (LPL) and three anonymous reviewers for their comments on the previous versions of the paper.

References

R. Baeza-Yates, L. Rello, and J. Dembowski. 2015. Cassa: a context-aware synonym simplification algorithm. In *Proceedings of the 2015 NAACL:HLT Conference*, pages 1380–1385.

D. Bernhard. 2006. Unsupervised morphological segmentation based on segment predictability and word segments alignment. In *Proceedings of 2nd Pascal Challenges Workshop*, pages 19–24.

D. Bernhard. 2010. Apprentissage non supervisé de familles morphologiques: Comparaison de méthodes et aspects multilingues. *Traitement Automatique des Langues*, 51(2):11–39.

M. B. Billami and N. Gala. 2017. Creating and validating semantic signatures : application for measuring semantic similarity and lexical substitution. In *Traitement Automatique des Langues Naturelles TALN 2017*, pages 123–138, Orléans, France, June.

A. Budanitsky and G. Hirst. 2006. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1):13–47, Mars. MIT Press.

M. Coltheart. 1978. Lexical access in simple reading tasks. In G. Underwood, editor, *Strategies of information processing*, pages 151–216. Academic Press, London.

Conseil de l'Europe. 2001. *Cadre européen commun de référence pour les langues : apprendre, enseigner, évaluer*. Hatier, Paris.

T. François, N. Gala, P. Watrin, and C. Fairon. 2014. FLELex: a graded lexical resource for French foreign learners. In *Proceedings of LREC 2014*, Reykjavik, Island.

N. Gala, T. François, and C. Fairon. 2013. Towards a French lexicon with difficulty measures: NLP helping to bridge the gap between traditional dictionaries and specialized lexicons. In *E-lexicography in the 21st century: thinking outside the paper*, Tallin, Estonie.

N. Gala, T. François, D. Bernhard, and C. Fairon. 2014. Un modèle pour prédire la complexité lexicale et graduer les mots. In *Actes de la 21e conférence sur le Traitement Automatique des Langues Naturelles (TALN'2014)*, pages 91–102.

N. Gala, M. B. Billami, T. François, and D. Bernhard. 2015. Graded lexicons: new resources for educational purposes and much more. In *22nd Computer-assisted language learning conference (EUROCALL-2015)*, pages 204–209, Padoue, Italie, August.

G. Gougenheim. 1958. *Dictionnaire fondamental de la langue française*. Didier, Paris.

R. Herbrich, T. Graepel, and K. Obermayer. 2000. Large margin rank boundaries for ordinal regression. chapter 7, pages 115–132. MIT Press, Cambridge.

M. Lafourcade. 2007. Making people play for Lexical Acquisition with the JeuxDeMots prototype. In *SNLP'07: 7th International Symposium on NLP*, Pattaya, Chonburi, Thailande.

B. Lété, L. Sprenger-Charolles, and P. Colé. 2004. Manulex : A grade-level lexical database from French elementary-school readers. *Behavior Research Methods, Instruments and Computers*, 36:156–166.

G. A. Miller, R. Beckwith, Ch. Fellbaum, D. Gross, and K. Miller. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3:235–244.

M. Morel and J. François. 2015. Le Dictionnaire Electronique des Synonymes du CRISCO : un outil de plus en plus interactif. *Revue française de linguistique appliquée*, XX(01):9–28.

R. Navigli and S. P. Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

R. Navigli. 2009. Word Sense Disambiguation: a survey. *ACM Computing Surveys*, 41(2):1–69.

B. New, M. Brysbaert, J. Veronis, and C. Pallier. 2007. The use of film subtitles to estimate word frequencies. *Applied Psycholinguistics*, 28(04):661–677.

C. K. Ogden. 1930. *Basic English*. Paul Treber, London.

A. Polguère. 2002. *Notions de base en lexicologie*. Observatoire de Linguistique Sens-Texte, Université de Montréal, Montréal.

B. Sagot and D. Fiser. 2008. Building a free French wordnet from multilingual resources. In *Ontolex*, Marrakech.

L. Specia, Sujay K. Jauhar, and R. Mihalcea. 2012. Semeval-2012 task 1: English lexical simplification. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012)*, Montréal.

K. Tanaka-Ishii, S. Tezuka, and H. Terada. 2010. Sorting texts by readability. *Computational Linguistics*, 36(2):203–227.

J. Tharp. 1939. The measurement of vocabulary difficulty. *Modern Language Journal*, pages 169–178.

E. Thorndike. 1921. *The Teacher's Word Book*. Teachers College, Columbia University, New York.

Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188, Janvier. AI Access Foundation.

Assaf Urieli. 2013. *Robust French syntax analysis: reconciling statistical methods and linguistic knowledge in the Talismane toolkit*. Ph.D. thesis, University of Toulouse II le Mirail, France.

Appendix A. List of synonyms evaluated (ranked) by human judges.

The following table shows the list of synonyms provided to the human judges for manual ranking. The words where presented decontextualized, randomly organized into a list correspondig to the same meaning. The judges had to rank them from 1 to n according to the difficulty they had to read and understand them. The table presents one line per vector of synonyms with an English translation at the end.

associer	combiner	assimiler	entrêmeler	amalgamer	to blend
bleu	azur	céruléen			blue
bleu	fromage				blue cheese
bleu	contusion	ecchymose			bruise
bleu	bizut	débutant	béjaune		beginner
brûler	cramer	incendier	cautériser	incinérer	to burn
intellectuel	cérébral				thinker
chic	élégant	huppé	aristocrate		elegant
agent	gendarme	connétable	agent de police		policemen
conte	fable	allégorie	apologue	histoire	tale
conte	narration				story
proximité	voisinage	contiguïté			nearness
noble	généreux	galant	héroïque	chevaleresque	gentle
dépouiller	apercevoir	constater	déceler	analyser	to notice
voler	piquer	dépouiller	dérober		to steal
inventer	forger	formuler			to invent
murmure	bruissement	gazouillis	gazouillement		whisper
pardon	grâce	amnistie	droit de grâce	grâce présidentielle	forgiveness
injure	affront	insulte			insult
mine	galerie	gissement	excavation	creusement	mine
mine	puits	charbonnage	huillère		pit
mine	plomb	mine de crayon			pencil lead
mine	gueule	galibot			expression
mine	mine antichar	mine antipersonnel			landmine
air	mine	manière	présence	comportement	face
parfois	tantôt	quelquefois	occasionnellement		sometimes
mémoire	rappel	réminiscence			memory
rappel	descente en rappel				abseiling
rougir	empourprer	cramoisir			to blush
fin	spirituel	mental			mental
merveilleux	fantastique	fabuleux	formidable	splendide	wonderful
maigre	osseux	squelettique			thin
sévère	rigoureux	strict	austère		strict
gémir	rugir	vagir			to roar
crier	hurler	brailler	beugler	vociférer	to yell
rugir	ronfler	bourdonner	vrombir		to hum

Table 6: ReSyf data evaluated by 40 human judges.