Automated WordNet Construction Using Word Embeddings

Mikhail Khodak, Andrej Risteski, Christiane Fellbaum and Sanjeev Arora

Computer Science Department, Princeton University, 35 Olden St., Princeton, New Jersey 08540

{mkhodak,risteski,fellbaum,arora}@cs.princeton.edu

Abstract

We present a fully unsupervised method for automated construction of WordNets based upon recent advances in distributional representations of sentences and word-senses combined with readily available machine translation tools. The approach requires very few linguistic resources and is thus extensible to multiple target languages. To evaluate our method we construct two 600-word test sets for word-to-synset matching in French and Russian using native speakers and evaluate the performance of our method along with several other recent approaches. Our method exceeds the best language-specific and multi-lingual automated WordNets in F-score for both languages. The databases we construct for French and Russian, both languages without large publicly available manually constructed WordNets, will be publicly released along with the test sets.

1 Introduction

A *WordNet* is a lexical database for languages based upon a structure introduced by the Princeton WordNet (PWN) for English in which sets of cognitive synonyms, or synsets, are interconnected with arcs standing for semantic and lexical relations between them (Fellbaum, 1972). WordNets are widely used in computational linguistics, information retrieval, and machine translation. Constructing one by hand is time-consuming and difficult, motivating a search for automated or semi-automated methods. We present an unsupervised method based on word embeddings and word-sense induction and build and evaluate WordNets for French and Russian. Our approach needs only a large unannotated corpus like Wikipedia in the target language and machine translation (MT) between that language and English.

A standard minimal WordNet design is to have synsets connected by hyponym-hypernym relations and linked back to PWN (Global WordNet Association, 2017). This allows for applications to cross-lingual tasks and rests on the assumption that synsets and their relations are invariant across languages (Sagot and Fišer, 2008). For example, while all senses of English word *tie* may not line up with all senses of French word *cravate*, the sense "necktie" will exist in both languages and be represented by the same synset.

Thus MT is often used for automated WordNets to generate a set of candidate synsets for each word w in the target language by getting a set of English translations of w and using them to query PWN (we will refer to this as MT+PWN). The number of candidate synsets produced may not be small, even as large as a hundred for some polysemous verbs. Thus one needs a way to select from the candidates of w those synsets that are its true senses. The main contributions of this paper is a new *word embedding*-based method for matching words to synsets and the release of two large word-synset matching test sets for French and Russian.[1]

Though there has been some work using word-vectors for WordNets (see Section 2), the resulting databases have been small, containing less than 1000 words. Using embeddings for this task is challenging due to the need for good ways to use PWN synset information and account for the breakdown of cosine-similarity for polysemous words. We approach the first issue by representing synset information using recent work on sentence-embeddings by Arora et al. (2017). To handle polysemy we devise a sense clustering scheme based on Word Sense Induction (WSI) via linear alge-

[1] https://github.com/mkhodak/pawn

Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications, pages 12–23,
Valencia, Spain, April 4 2017.

bra over word-vectors (Arora et al., 2016a). We demonstrate how this *sense purification* procedure effectively combines clustering with embeddings, thus being applicable to many word-sense disambiguation (WSD) and induction-related tasks. Using both techniques yields a WordNet method that outperforms other language-independent methods as well as language-specific approaches such as WOLF, the French WordNet used by the Natural Language ToolKit (Sagot and Fišer, 2008; Bond and Foster, 2013; Bird et al., 2009).

Our second contribution is the creation of two new 600-word test sets in French and Russian that are larger and more comprehensive than any currently available, containing 200 each of nouns, verbs, and adjectives. They are constructed by presenting native speakers with all candidate synsets produced as above by MT+PWN and treating the senses they pick out as "ground truth" for measuring precision and recall. The motivation behind separating by part-of-speech (POS) is that nouns are often easier than adjectives and verbs, so reporting one number — as done by some past work — allows high noun performance to mask low performance on adjectives and verbs.

Using these test sets, we can begin addressing the difficulties of evaluation for non-English automated WordNets due to the use of different and unreported test data, incompatible metrics (e.g. matching synsets to words vs. retrieving words for synsets), and differing cross-lingual dictionaries. In this paper we use the test sets to evaluate our method and several other automated WordNets.

2 Related Work

There have been many language-specific approaches for building automated WordNets, notably for Korean (Lee et al., 2000), French (Sagot and Fišer, 2008; Pradet et al., 2013), and Persian (Montazery and Faili, 2010). These approaches also use MT+PWN to get candidate word-synset pairs, but often use further resources — such as bilingual corpora, expert knowledge, or WordNets in related languages — to select correct senses.

The Korean construction depends on a classifier trained on 3260 word-sense matchings that yields 93.6% precision and 77.1% recall, albeit only on nouns. The Persian WordNet uses a scoring function based on related words between languages (requiring expert knowledge and parallel corpora) and achieves 82.6% precision, though without reporting recall and POS-separated statistics.

The most comparable results to ours are from the Wordnet Libre du Français (WOLF) of Sagot and Fišer (2008), which leverages multiple European WordNet projects. Our best method exceeds this approach on our test set and benefits from having far fewer resource requirements. The Wordnet du Français (WoNeF) of Pradet et al. (2013) depends on combining linguistic models by a voting scheme. Their performance is found to be generally below WOLF's, so we compare to the latter.

There has also been work on multi-language WordNets, specifically the Extended Open Multilingual Wordnet (OMW) (Bond and Foster, 2013), which scraped Wiktionary, and the Universal Multilingual Wordnet (UWN) (de Melo and Weikum, 2009), which used multiple translations to rate word-sense matches. In our evaluation both produce high-precision/low-coverage WordNets.

Finally, there have been recent vector approaches for an Arabic WordNet (Tarouti and Kalita, 2016) and a Bengali WordNet (Nasiruddin et al., 2014). The Arabic effort uses a cosine-similarity threshold for correcting direct translation and reports a precision of 78.4% on synonym matching, although its small size (943 synsets) indicates a poor precision/recall trade-off. The Bengali WordNet paper examines WSI on word vectors, evaluating clustering methods on seven words and achieving F_1-scores of at best 52.1%. It is likely that standard clustering techniques are insufficient when one needs many thousands of clusters, an issue we address via sparse coding.

Our use of distributional word embeddings to construct WordNets is the latest in a long line of their applications, e.g. approximating word similarity and solving word-analogies (Mikolov et al., 2013). The latter discovery was cited as the inspiration for the theoretical model in Arora et al. (2016b), whose Squared-Norm (SN) vectors we use; the computation is similar in form and performance to GloVe (Pennington et al., 2014).

3 Methods for WordNet Construction

The basic WordNet method is as follows. Given a target word w, we use a bilingual dictionary to get its translations in English and let its set of candidate synsets be all PWN senses of the translations (MT+PWN). We then assign a score to each synset and accept as correct all synsets with score above a threshold α If no synset is above the cutoff we

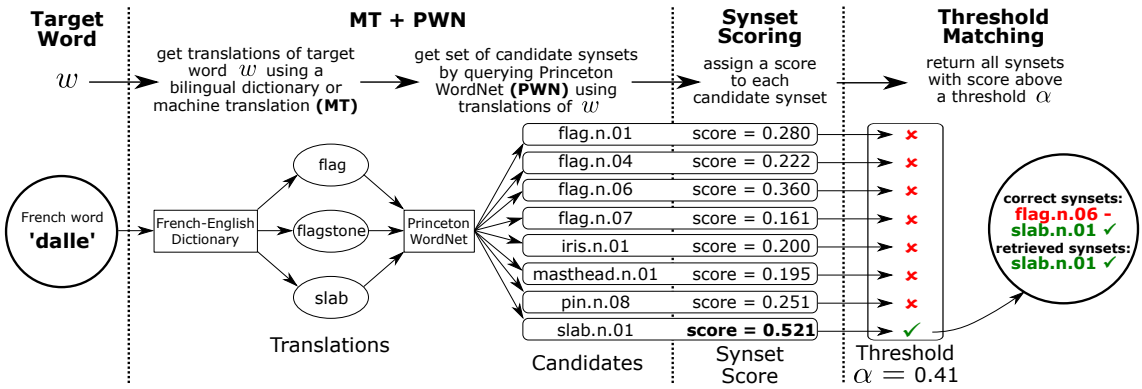

Figure 1: The score-threshold procedure for French word $w = \textit{dalle}$ (flagstone, slab). Candidate synsets generated by MT+PWN are given a score and matched to w if the score is above a threshold α.

accept only the highest-scoring synset. This *score-threshold* method is illustrated in Figure 1.

Thus we need methods to assign high scores to correct candidate synsets of w and low scores to incorrect ones. We use unsupervised word vectors in the target language computed from a large text corpus (e.g. Wikipedia). Section 3.1 presents a simple baseline that improves upon MT+PWN via a cosine-similarity metric between w and each synset's lemmas. A more sophisticated *synset representation* method using sentence-embeddings is described in Section 3.2. Finally, in Section 3.3 we discuss WSI-based procedures for improving the score-threshold method for words with fine sense distinctions or poorly-annotated synsets.

In this section we assume a vocabulary V of target language words with associated d-dimensional unit word-vectors $v_w \in \mathbb{R}^d$ for $d \ll |V|$ (e.g. $d = 300$ for vocabulary size 50000) trained on a large text corpus. Each word $w \in V$ also has a set of candidate synsets found by MT+PWN. We call synsets S, S' in PWN *related*, denoted $S \sim S'$, if one is a hyponym, meronym, antonym, or attribute of the other, if they share a verb group, or $S = S'$.

3.1 Baseline: Average Similarity Method

This method for scoring synsets can be seen as a simple baseline. Given a candidate synset S, we define $T_S \subset V$ as the set of translations of its lemmas from English to the target language. The score of S is then $\frac{1}{|T_S|} \sum_{w' \in T_S} v_w \cdot v_{w'}$, the average cosine similarity between w and the translated lemmas of S. Although straightforward, this scoring method is quite noisy, as averaging word-vectors dilutes similarity performance, and does not use all synset information provided by PWN.

3.2 Method 1: Synset Representation

To improve upon this baseline we need a better vector representation of S to score S via cosine similarity with v_w. Previous efforts in synset and sense embeddings (Iacobacci et al., 2015; Rothe and Schütze, 2015) often use extra resources such as WordNet or BabelNet for the target language (Navigli and Ponzetto, 2012). As such databases are not always available, we propose a synset representation u_S that is unsupervised, needing no extra resources beyond MT and PWN, and leverages recent work on sentence embeddings.

This new representation combines embeddings of synset information given by PWN, e.g. synset relations, definitions, and example sentences. To create these embeddings we first consider the question of how to represent a list of words L as a vector in $\mathbb{R}^d$. One way is to simply take the normalized sum $\hat{v}_L^{(SUM)}$ of their word-vectors, where

$$v_L^{(SUM)} = \sum_{w' \in L} v_{w'}$$

Potentially more useful is to compute a vector $\hat{v}_L^{(SIF)}$ via the sentence embedding formula of Arora et al. (2017), based on *smooth inverse frequency* (SIF) weighting, which (for $a = 10^{-4}$ and before normalization) is expressed as

$$v_L^{(SIF)} = \sum_{w' \in L} \frac{a}{a + \mathbb{P}\{w'\}} v_{w'}$$

SIF is similar in spirit to TF-IDF (Salton and Buckley, 1988) and builds on work of Wieting et al. (2016); it has been found to perform well on other similarity tasks (Arora et al., 2017).

We find that SIF improves performance on sentences but not on translated lemma lists (Figure 2),

likely because sentences contain many distractor words that SIF will weight lower while the presence of distractors among lemmas is independent of word frequency. Thus to compute the synset score $u_S \cdot v_w$ we make the vector representation u_S of S the element-wise average of:

- $\hat{v}^{(SUM)}_{T_S}$ — T_S is the set of translations of lemmas of S (as in Section 3.1).
- $\hat{v}^{(SUM)}_{R_S}$ — $R_S = \left(\bigcup_{S' \sim S} T_{S'} \right) \setminus T_S$ is the set of lemma translations of synsets S' related to S.
- $\hat{v}^{(SIF)}_{D_S}$ — D_S is the list of tokens in the translated definition of S.
- $\frac{1}{|\mathcal{E}_s|} \sum_{E \in \mathcal{E}_S} \hat{v}^{(SIF)}_E$ — $\mathcal{E}_S$ contains the lists of tokens in the translated example sentences of S (excluded if S has no example sentences).

3.3 Method 2: Better Matching Using WSI

We found through examination that the score-threshold method we have used so far performs poorly in two main cases:

(a) the word w has no candidate synset with score high-enough to clear the threshold.

(b) w has multiple closely related synsets that are all correct matches but some of which have a much lower score than others.

Here we address both issues by using sense information found by applying a word-sense induction method first introduced in Arora et al. (2016a).

We summarize their WSI-model – referred to henceforth as *Linear-WSI* — in Section 3.3.1. Then in Section 3.3.2 we devise a *sense purification* procedure for constructing a word-cluster for each induced sense of a word. Applying this procedure to construct word-cluster representations of candidates synsets provides an additional metric for the correctness of word-synset matches that can be used to devise a w-specific threshold α_w to ameliorate problem (a). Meanwhile, using Linear-WSI to associate similar candidate synsets of w to each other provides a way to address problem (b). We explain these methodologies in Section 3.3.3.

3.3.1 Summary of Linear-WSI Model

In Arora et al. (2016a) the authors posit that the vector of a polysemous word can be linearly decomposed into vectors associated to its senses.

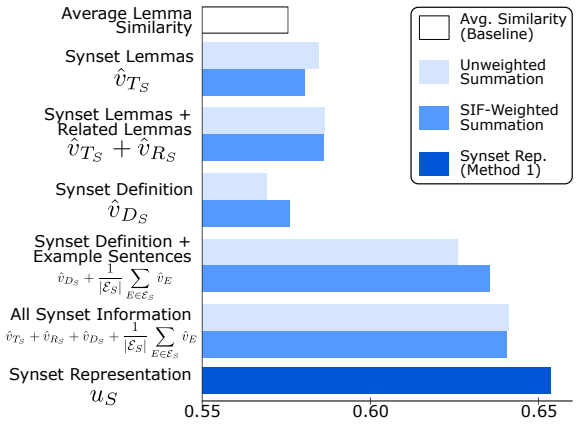

Figure 2: F-score comparison between using unweighted summation and SIF-weighted summation for embedding PWN synset information.

Thus for $w = tie$ — which can be an article of clothing, a drawn match, and so on — we would have $v_w \approx a v_{w(clothing)} + b v_{w(match)} + \ldots$ for $a, b \in \mathbb{R}$. It is unclear how to find such sense-vectors, but one expects different words to have closely related sense-vectors, e.g. for $w' = bow$ the vector $v_{w'(clothing)}$ would be close to the vector $v_{w(clothing)}$ of *tie*. Thus the Linear-WSI model proposes using sparse coding, namely finding a set of unit basis vectors $a_1, \ldots, a_k \in \mathbb{R}^d$ s.t. $\forall\, w \in V$,

$$v_w = \sum_{i=1}^{k} R_{wi} a_i + \eta_w, \qquad (1)$$

for $k > d$, η_w a noise vector, and at most s coefficients R_{wi} nonzero. The hope is that the sense-vector $v_{w(clothing)}$ of *tie* is in the neighborhood of a vector a_i s.t. $R_{wi} > 0$. Indeed, for $k = 2000$ and $s = 5$, Arora et al. (2016a) report that solving (1) represents English word-senses as well as a competent non-native speaker and significantly better than older clustering methods for WSI.

3.3.2 Sense Purification

While (1) is a good WSI method, its use for building WordNets is hampered by its inability to produce more than a few thousand senses a_i, as setting a large k yields repetitive rather than different senses. As this is far fewer than the number of synsets in PWN, we use sense purification to address this by extracting a cluster of words related to a_i as well as w to represent each sense. In addition to w and a_i, the procedure takes as input a search-space $V' \subset V$ and set size n. Then we find

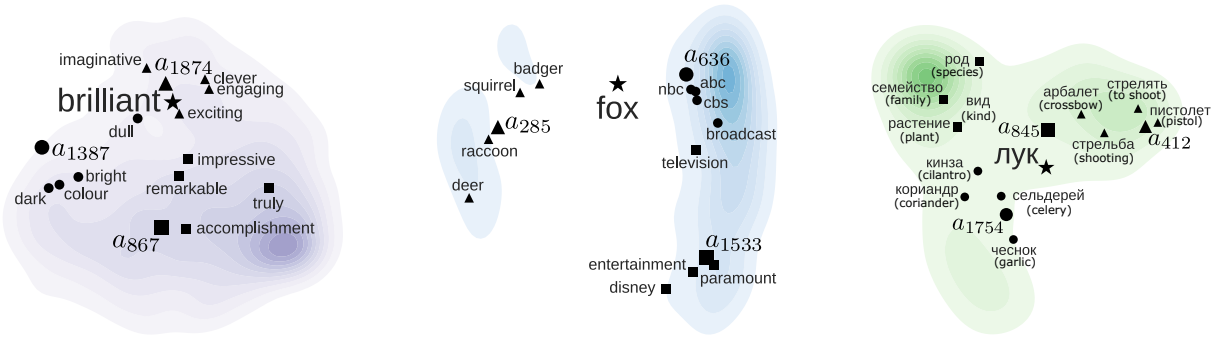

Figure 3: Isometric mapping of sense-cluster vectors for $w =$ *brilliant*, *fox*, and лук (bow, onion). w is marked by a star and each sense a_i of w, shown by a large marker, has an associated cluster of words with the same marker shape. Contours are densities of vectors close to w and at least one sense a_i.

$C \subset V'$ of size n containing w that maximizes

$$f = \min_{\substack{x=a_i \text{ or} \\ x=v_{w'}:w' \in C}} \text{Median}\{x \cdot v_{w'} : w' \in C\} \quad (2)$$

A cluster C maximizing this must ensure that neither a_i nor any word $w' \in C$ (including $w' = w$) has low average cosine similarity with cluster-words, resulting in a dense cluster close to both w and a_i. We explain this further in Appendix A.

We illustrate this method in Figure 3 by purifying the senses of English words *brilliant* and *fox* and Russian word лук, which has the senses "bow" (weapon), "onion" (vegetable), and "onion" (plant). Note how correct senses are recovered across POS and language and for proper and common noun senses.

3.3.3 Applying WSI to Synset Matching

The problem addressed by sense purification is that senses a_i induced by Linear-WSI have too many related words; purification solves this by extracting a cluster of words related to w from the words close to a_i. When translating WordNet synsets, we have a similar problem in that translations of a synset's lemmas may not be relevant to the synset itself. Thus we can try to create a purified representation of each candidate synset S of w by extracting a cluster of translated lemmas close to w and one of its induced senses. We run purification on every sense a_i in the sparse representation (1) of w, using as a search-space $V' = V_S$ the set of translations of all lemmas of synsets S' related to S in PWN (i.e. $S' \sim S$ as defined in Section 3). To each synset S we associate the sense a_S and corresponding cluster C_S that are optimal in the objective (2) among all senses a_i of w.

Although we find a sense a_S and purified representation C_S for each candidate synset of w, we note that an incorrect synset is likely to have a lower objective value (2) than a correct synset as it likely has fewer words related to w in its search-space V_S. However, using $f_S = f(w, a_S, C_S)$ as a synset score is difficult as some synsets have very small search-spaces, leading to inconsistent scoring even for correct synsets.

Instead we use f_S as part of a w-specific threshold $\alpha_w = \min\{\alpha, u_{S^*} \cdot v_w\}$, where u_S is the vector representation of S from Section 3.2 and $S^* = \arg\max f_S + u_S \cdot v_w$. This attempts to address problem (a) of the score-threshold method — that some words have no synsets above the cutoff α and returning only the highest-scoring synset in such cases will not retrieve multiple sense for polysemous words. By the construction of α_w, if w is polysemous, a synset other than the one with the highest score $u_S \cdot v_w$ may have a better value of f_S and thus found to be S^*; then all synsets with higher scores will be matched to w, allowing for multiple matches even if no synset clears the cutoff α. Otherwise if w is monosemous, we expect the correct synset S to have the highest value for both $u_S \cdot v_w$ and f_S, making $S^* = S$. Then if no synset has score greater than α the threshold will be set to $u_S \cdot v_w$, the highest synset-score, so only the highest scoring synset will be matched to w.

To address problem (b) of the threshold method — that w might have multiple correct and closely related candidate synsets of which only some clear the cutoff — we observe that closely related synsets of w will have similar search-spaces V_S and so are likely to be associated to the same sense a_i. For example, in Figure 4 the candidates of *dalle* related to its correct meanings as a flagstone or slab are associated to the same sense a_{892} while distractor synsets related to the incorrect sense as a

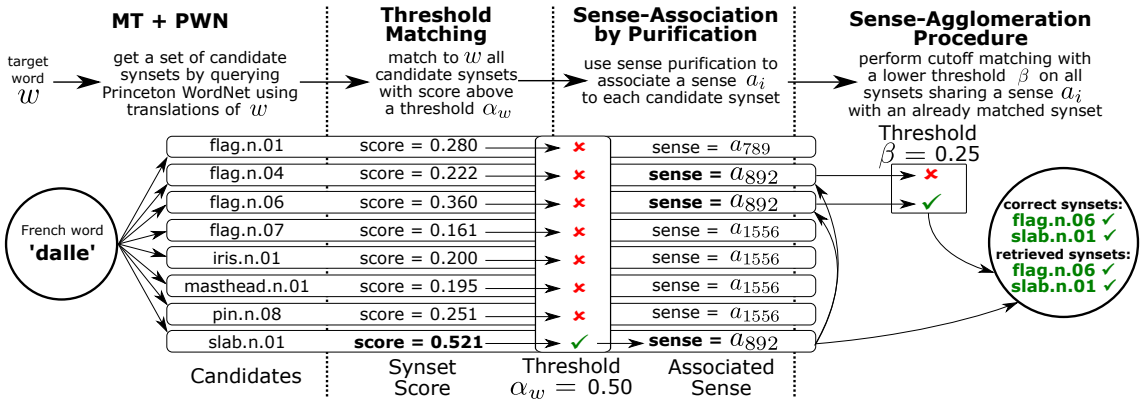

Figure 4: The score-threshold and sense-agglomeration procedure for French word $w =$ *dalle* (flagstone, slab). Candidate synsets are given a score and matched to w if they clear a high threshold α_w (as in Section 3.2). If an unmatched synset shares a sense a_i with a matched synset, it is compared to a low threshold β (the sense-agglomeration procedure in Section 3.3.3).

flag are mostly matched to other senses. This motivates the *sense agglomeration* procedure, which uses threshold-clearing synsets to match synsets with the same sense below the score-threshold. For $\beta < \alpha$, the procedure is roughly as follows:

1. Run score-threshold method with cutoff α_w.

2. For each synset S with score $u_S \cdot v_w$ below the threshold, check for a synset S' with score $u_{S'} \cdot v_w$ above the threshold and $a_{S'} = a_S$.

3. If $u_S \cdot v_w \geq \beta$ and the clusters $C_S, C_{S'}$ satisfy a cluster similarity condition, match S to w.

We include the lower cutoff β because even similar synsets may not both be senses of the same word. The cluster similarity condition, available in Appendix B.2, ensures the relatedness of synsets sharing a sense a_i, as an erroneous synset S' may be associated to the same sense as a correct one.

In summary, the improved synset matching approach has two steps: 1-conduct score-threshold matching using the modified threshold α_w; 2-run the sense-agglomeration procedure on all senses a_i of w having at least one threshold-clearing synset S. Although for fixed α both steps focus on improving the recall of the threshold method, in practice they allow α to be higher, so that both precision and recall are improved. For example, note the recovery of a correct synset left unmatched by the score-threshold method in the simplified depiction of sense-agglomeration shown in Figure 4.

4 Evaluation of Methods

We evaluate our method's accuracy and coverage by constructing and testing WordNets for French and Russian. For both we train 300-dimensional SN word embeddings (Arora et al., 2016b) on co-occurrences of words occurring at least 1000 times, or having candidate PWN synsets and occurring at least 100 times, in the lemmatized Wikipedia corpus. This yields $|V| \approx 50000$. For Linear-WSI we run sparse coding with sparsity $s = 4$ and basis-size $k = 2000$ and use set-size $n = 5$ for purification. To get candidate synsets we use Google and Microsoft Translators and the dictionary of the translation company ECTACO, while for sentence-length MT we use Microsoft.

4.1 Testsets

A common way to evaluate accuracy of an automated WordNet is to compare its synsets or word matchings to a manually-constructed one. However, the existing ELRA French Wordnet [2] is not public and half the size of ours while Russian WordNets are either even smaller and not linked to PWN[3] or obtained via direct translation[4].

Instead we construct test sets for each language that allow for evaluation of our methods and others. We randomly chose 200 each of adjectives, nouns, and verbs from the set of target language words whose English translations appear in the synsets of the Core WordNet. Their "ground truth"

[2] http://catalog.elra.info/product_info.php?products_id=550

[3] http://project.phil.spbu.ru/RussNet/

[4] http://wordnet.ru/

Method	POS	$F_{.5}$-Score*	F_1-Score*	Precision*	Recall*	Coverage	Synsets	α	β
Direct Translation (MT + PWN)	Adj.	50.3	59.3	46.1	100.0	99.9	11271		
	Noun	56.2	64.6	52.2	100.0	100.0	74477		
	Verb	41.4	51.3	37.0	100.0	100.0	13017		
	Total	49.3	58.4	45.1	100.0	100.0	100174†		
Wordnet Libre du Français (WOLF) (Sagot and Fišer, 2008)	Adj.	66.3	58.6	78.1	53.4	84.8	6865		
	Noun	68.6	58.7	83.2	51.5	95.0	36667		
	Verb	60.8	48.4	81.0	39.6	88.2	7671		
	Total	65.2	55.2	80.8	48.2	92.2	52757†		
Universal Wordnet (de Melo and Weikum, 2009)	Adj.	64.5	51.5	88.3	42.3	69.2	7407		
	Noun	67.5	52.2	94.1	40.8	75.9	24670		
	Verb	55.4	39.5	88.0	28.5	76.2	5624		
	Total	62.5	47.7	90.1	37.2	75.0	39497†		
Extended Open Multilingual Wordnet (Bond and Foster, 2013)	Adj.	58.4	40.8	**90.9**	28.4	54.7	2689		
	Noun	61.3	43.8	**96.5**	31.7	66.6	14936		
	Verb	47.8	29.4	**95.9**	18.6	57.7	2331		
	Total	55.9	38.0	**94.5**	26.2	63.2	20449†		
Baseline: Average Similarity (Section 3.1)	Adj.	62.8±0.0	62.6±0.0	65.3±0.0	**68.5**±0.0	88.7	9687	0.31	
	Noun	67.3±0.0	65.4±0.1	71.6±0.1	**69.0**±0.1	92.2	37970	0.27	
	Verb	51.8±0.0	50.8±0.1	55.9±0.1	**57.0**±0.1	83.5	10037	0.26	
	Total	60.6±0.0	59.6±0.0	64.3±0.0	**64.9**±0.0	90.0	58962†		
Method 1: Synset Representation (Section 3.2)	Adj.	65.9±0.0	60.4±0.0	75.9±0.1	59.5±0.1	85.1	8512	0.47	
	Noun	71.0±0.0	**67.3**±0.1	78.7±0.1	69.1±0.1	**96.7**	35663	0.41	
	Verb	61.6±0.0	53.0±0.0	78.7±0.1	49.8±0.1	89.9	8619	0.45	
	Total	66.2±0.0	60.2±0.0	77.8±0.0	59.5±0.1	**93.7**	53852†		
Method 2: Synset Representation + Linear-WSI (Section 3.3)	Adj.	**67.7**±0.0	**62.5**±0.1	76.9±0.1	62.6±0.1	**91.2**	8912	0.56	0.42
	Noun	**73.0**±0.0	66.0±0.1	83.7±0.1	62.0±0.2	90.9	34001	0.50	0.25
	Verb	**64.4**±0.0	**55.9**±0.0	79.3±0.0	51.5±0.1	**93.6**	9262	0.46	0.28
	Total	**68.4**±0.0	**61.5**±0.0	80.0±0.0	58.7±0.1	91.5	53208†		

* Micro-averages over a randomly held-out half of the data; parameters tuned on the other half. 95% asymptotic confidence intervals found with 10000 randomized trials.

† Includes adverb synsets. For the last three methods they are matched with the same parameter values (α and β) as for adjectives.

Table 1: French WordNet Results

word senses are picked by native speakers, who were asked to perform the same matching task described in Section 3, i.e. select correct synsets for a word given a set of candidates generated by MT + PWN. For example, the French word *foie* has one translation, *liver* with four PWN synsets: 1-"glandular organ"; 2-"liver used as meat"; 3-"person with a special life style"; 4-"someone living in a place." The first two align with senses of *foie* while the others do not, so the expert marks the first two as good and the others as negative. Two native speakers for each language were trained by an author with knowledge of WordNet and at least 10 years of experience in each language. Inconsistencies in the matchings of the two speakers were resolved by the same author.

We get 600 words and about 12000 candidate word-synset pairs for each language, with adjectives and nouns having on average about 15 candidates and verbs having about 30. These numbers makes the test set larger than many others, with French, Korean, and Persian WordNets cited in Section 2 being evaluated on 183 pairs, 3260 pairs, and 500 words, respectively. Accuracy measured with respect to this ground truth estimates how well an algorithm does compared to humans.

A significant characteristic of this test set is its dependence on the machine translation system used to get candidate synsets. While this can leave out correct synset matches that the system did not propose, by providing both correct and incorrect candidate synsets we allow future authors to focus on the semantic challenge of selecting correct senses without worrying about finding the best bilingual dictionary. This allows dictionary-independent evaluation of automated WordNets, an important feature in an area where the specific translation systems used are rarely provided in full. When comparing the performance of our construction to that of previous efforts on this test set, we do not penalize word-synset matches in which the synset is not among the candidate synsets we

generate for that word, negating the loss of precision incurred by other methods due to the use of different dictionaries. We also do not penalize other WordNets for test words they do not contain.

In addition to precision and recall, we report the *Coverage* statistic as the proportion of the Core set of most-used synsets, a semi-automatically constructed set of about 5000 PWN frequent senses, that are matched to at least one word (Fellbaum, 1972). While an imperfect metric given different sense usage by language, the synsets are universal-enough for it to be a good indicator of usability.

4.2 Evaluation Results

We report the evaluation of methods in Section 3 in Tables 1 & 2 alongside evaluations of UWN (de Melo and Weikum, 2009), OWM (Bond and Foster, 2013), and WOLF (Sagot and Fišer, 2008). Parameters α and β were tuned to maximize the micro-averaged $F_{.5}$-score $\frac{1.25 \cdot \text{Precision} \cdot \text{Recall}}{.25 \cdot \text{Precision} + \text{Recall}}$, used instead of F_1 to prioritize precision, which is often more important for application purposes.

Our synset representation method (Section 3.2) exceeds the similarity baseline by 6% in $F_{.5}$-score for French and 10% for Russian. For French it is competitive with the best other WordNet (WOLF) and in both languages exceeds both multi-lingual WordNets. Improving this method via Linear-WSI (Section 3.3) leads to 2% improvement in $F_{.5}$-score for French and 1% for Russian. Our methods also perform best in F_1-score and Core coverage.

As expected from a Wiktionary-scraping method, OMW achieves the best precision across languages, although it and UWN have low recall and Core coverage. The performance of our best method for French exceeds that of WOLF in $F_{.5}$-score across POS while achieving similar coverage. WOLF's recall performance is markedly lower than the evaluation in Sagot and Fišer (2008, Table 4); we believe this stems from our use of words matched to Core synsets, not random words, leading to a more difficult test set as common words are more-polysemous and have more synsets to retrieve. There is no comparable automated Russian-only WordNet, with only semi-automated and incomplete efforts (Yablonsky and Sukhonogov, 2006).

Comparing across POS, we do best on nouns and worst on verbs, likely due to the greater polysemy of verbs. Between languages, performance is similar for adjectives but slightly worse on Russian nouns and much worse on Russian verbs.

The discrepancy in verbs can be explained by a difference in treating the reflexive case and aspectual variants due to the grammatical complexity of Russian verbs. In French, making a verb reflexive requires adding a word while in Russian the verb itself changes, e.g. *to wash→to wash oneself* is *laver→se laver* in French but мыть→мыться in Russian. Thus we do not distinguish the reflexive case for French as the token found is the same but for Russian we do, so both мыть and мыться may appear and have distinct synset matches. Thus matching Russian verbs is challenging as the reflexive usage of a verb is often contextually similar to the non-reflexive usage. Another complication for Russian verbs is due to aspectual verb pairs; thus *to do* has aspects (делать, сделать) in Russian that are treated as distinct verbs while in French these are just different tenses of the verb *faire*. Both factors pose challenges for differentiating Russian verb senses by a distributional model.

Overall however the method is shown to be robust to how close the target language is to English, with nouns and adjectives performing well in both languages and the difference for verbs stemming from an intrinsic quality rather than dissimilarity with English. This can be further examined by testing the method on a non-European language.

5 Conclusion and Future Work

We have shown how to leverage recent advances in word embeddings for fully-automated WordNet construction. Our best approach combining sentence embeddings, and recent methods for WSI obtains performance 5-16% above the naive baseline in $F_{.5}$-score as well as outperforming previous language-specific and multi-lingual methods. A notable feature of our work is that we require only a large corpus in the target language and automated translation into/from English, both available for many languages lacking good WordNets.

We further contribute new 600-word human-annotated test sets split by POS for French and Russian that can be used to evaluate future automated WordNets. These larger test sets give a more accurate picture of a construction's strengths and weaknesses, revealing some limitations of past methods. With WordNets in French and Russian largely automated or incomplete, the WordNets we build also add an important tool for multilingual natural language processing.

Method	POS	$F_{.5}$-Score*	F_1-Score*	Precision*	Recall*	Coverage	Synsets	α	β
Direct Translation (MT + PWN)	Adj.	50.2	59.6	45.9	100.0	99.6	11412		
	Noun	41.2	50.2	37.1	100.0	100.0	73328		
	Verb	32.5	41.7	28.6	100.0	100.0	13185		
	Total	41.3	50.5	37.2	100.0	99.9	99470†		
Universal Wordnet (de Melo and Weikum, 2009)	Adj.	52.4	38.8	80.3	29.6	51.0	11412		
	Noun	65.0	53.0	87.5	45.1	71.1	19564		
	Verb	48.1	34.8	74.8	25.7	65.0	3981		
	Total	55.1	42.2	80.8	33.4	67.1	30015†		
Extended Open Multilingual Wordnet (Bond and Foster, 2013)	Adj.	58.7	41.3	**91.7**	29.2	55.3	2419		
	Noun	67.8	53.1	**93.5**	42.5	68.4	14968		
	Verb	51.1	34.8	**84.5**	23.9	56.6	2218		
	Total	59.2	43.1	**89.9**	31.9	64.2	19983†		
Baseline: Average Similarity (Section 3.1)	Adj.	61.4±0.0	64.6±0.1	60.9±0.0	**77.3**±0.1	92.1	10293	0.24	
	Noun	55.9±0.0	54.8±0.1	59.9±0.1	59.9±0.1	77.0	32919	0.29	
	Verb	46.3±0.0	46.5±0.1	49.0±0.1	**55.1**±0.1	84.1	9749	0.21	
	Total	54.5±0.0	55.3±0.0	56.6±0.0	**64.1**±0.1	80.5	54372†		
Method 1: Synset Representation (Section 3.2)	Adj.	69.5±0.0	64.1±0.0	78.1±0.0	61.7±0.1	84.2	8393	0.43	
	Noun	69.8±0.0	65.5±0.0	77.6±0.1	66.0±0.1	85.2	29076	0.46	
	Verb	54.2±0.0	**51.1**±0.1	63.3±0.1	57.4±0.1	91.2	8303	0.39	
	Total	64.5±0.0	60.2±0.0	73.0±0.0	61.7±0.1	86.3	46911†		
Method 2: Synset Representation + Linear-WSI (Section 3.3)	Adj.	**69.7**±0.0	**64.9**±0.1	77.3±0.0	63.6±0.1	**93.3**	9359	0.43	0.35
	Noun	**71.6**±0.0	**67.6**±0.0	78.1±0.0	**68.0**±0.1	**91.0**	31699	0.46	0.33
	Verb	**54.4**±0.0	49.7±0.1	64.9±0.1	52.6±0.2	**91.9**	8582	0.44	0.33
	Total	**65.2**±0.0	**60.7**±0.0	73.4±0.0	61.4±0.1	**91.5**	50850†		

* Micro-averages over a randomly held-out half of the data; parameters tuned on the other half. 95% asymptotic confidence intervals found with 10000 randomized trials.

† Includes adverb synsets. For the last three methods they are matched with the same parameter values (α and β) as for adjectives.

Table 2: Russian WordNet Results

Further improvement to our work may come from other methods in word-embeddings, such as multi-lingual word-vectors (Faruqui and Dyer, 2014). Our techniques can also be combined with others, both language-specific and multi-lingual, for automated WordNet construction. In addition, our method for associating multiple synsets to the same sense can contribute to efforts to improve PWN through sense clustering (Snow et al., 2007). Finally, our sense purification procedure, which uses word-vectors to extract clusters representing word-senses, likely has further WSI and WSD applications; such exploration is left to future work.

Acknowledgments

We thank Angel Chang and Yingyu Liang for helpful discussion at various stages of this paper. This work was supported in part by NSF grants CCF-1302518, CCF-1527371, Simons Investigator Award, Simons Collaboration Grant, and ONR-N00014-16-1-2329.

References

Michal Aharon, Michael Elad, and Alfred Bruckstein. 2006. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11).

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016a. Linear algebraic structure of word sense, with applications to polysemy.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016b. Rand-walk: A latent variable model approach to word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*. To Appear.

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Francis Bond and Ryan Foster. 2013. Linking and extending an open multilingual wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

Gerard de Melo and Gerhard Weikum. 2009. Towards a universal wordnet by learning from combined evidence. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*.

Christiane Fellbaum. 1972. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Global WordNet Association. 2017. Wordnets in the world.

Ignaci Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*.

Changki Lee, Geunbae Lee, and Seo JungYun. 2000. Automated wordnet mapping using word sense disambiguation. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Mortaza Montazery and Heshaam Faili. 2010. Automatic persian wordnet construction. In *Proceedings of the 23rd International Conference on Computational Linguistics*.

Mohammad Nasiruddin, Didier Schwab, and Andon Tchechmedjiev. 2014. Induction de sens pour enrichir des ressources lexicales. In *21éme Traitement Automatique des Langues Naturelles*.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of Empirical Methods in Natural Language Processing*.

Quentin Pradet, Gaël de Chalendar, and Jeanne Baguenier Desormeaux. 2013. Wonef, an improved, expanded and evaluated automatic french translation of wordnet. In *Proceedings of the Seventh Global Wordnet Conference*.

Sascha Rothe and Hinrich Schütze. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*.

Benoît Sagot and Darja Fišer. 2008. Building a free french wordnet from multilingual resources. In *Proceedings of the Sixth International Language Resources and Evaluation Conference*.

Gerald Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5).

Rion Snow, Sushant Prakash, Daniel Jurafsky, and Andrew Y. Ng. 2007. Learning to merge word senses. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Feras Al Tarouti and Jugal Kalita. 2016. Enhancing automatic wordnet construction using word embeddings. In *Proceedings of the Workshop on Multilingual and Cross-lingual Methods in NLP*.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In *International Conference on Learning Representations*.

Sergey Yablonsky and Andrej Sukhonogov. 2006. Semi-automated english-russian wordnet construction: Initial resources, software and methods of translation. In *Proceedings of the Global WordNet Conference*.

A Purification Procedure

As discussed in Section 3.3.1, the Linear-WSI model (Arora et al., 2016a) posits that there exists an *overcomplete basis* $a_1, \ldots, a_k \in \mathbb{R}^d$ of unit vectors such that each word $w \in V$ can be approximately represented by a linear combination of at most s basis vectors (see Equation 1). Finding the basis vectors a_i and their coefficients R_{wi} requires solving the optimization problem

$$\begin{aligned} &\text{minimize} && \|Y - RA\|_2 \\ &\text{subject to} && \|R_w\|_0 \leq s \ \ \forall \ \ w \in V \end{aligned}$$

where $Y \in \mathbb{R}^{|V| \times n}$ has word-vectors as its rows, $R \in \mathbb{R}^{|V| \times k}$ is the matrix of coefficients R_{wi}, and $A \in \mathbb{R}^{k \times d}$ has the overcomplete basis $a_1, \ldots, a_k$ as its rows. This problem is non-convex and can be solved approximately via the K-SVD algorithm (Aharon et al., 2006).

Given a word w, the purification procedure is a method for purifying the senses induced via Linear-WSI (the vectors a_i s.t. $R_{wi} \neq 0$) by representing them as word-clusters related to both the sense itself and the word. This is done so as to create a more fine-grained collection of senses, as Linear-WSI does not perform well for more than a few thousand basis vectors, far fewer than the

number of word-senses (Arora et al., 2016a). The procedure is inspired by the hope that words related to each sense of w will form clusters near both v_w and one of its senses a_i. Given a fixed set-size n and a search-space $V' \subset V$, we realize this hope via the optimization problem

$$\begin{aligned}&\underset{C \subset V'}{\text{maximize}} \quad \gamma \\ &\text{subject to} \\ &\gamma \leq \text{Median}\{v_x \cdot v_{w'} : w' \in C \backslash \{x\}\} \ \forall \ x \in C \\ &\gamma \leq \text{Median}\{a_i \cdot v_{w'} : w' \in C\} \\ &w \in C, \ |C| = n\end{aligned}$$

This problem is equivalent to maximizing (2) with constraints $|C| = n$ and $w \in C \subset V'$. The objective value is constrained to be the lowest median cosine similarity between any word $x \in C$ or the sense a_i and the rest of C, so optimizing it ensures that the words in C are closely related to each other and to a_i. Forcing the cluster to contain w leads to the words in C being close to w as well.

For computational purposes we solve this problem approximately using a greedy algorithm that starts with $C = \{w\}$ and repeatedly adds the word in $V' \backslash C$ that results in the best objective value until $|C| = n$. Speed of computation is also a reason for using a search-space $V' \subset V$ rather than the entire vocabulary as a source of words for the cluster; we found that restricting V' to be all words w' s.t. $\min\{v_{w'} \cdot v_w, v_{w'} \cdot a_i\} \geq .2$ dramatically reduces processing time with little performance loss.

The purification procedure represents only one sense of w, so to perform WSI we generate clusters for all senses a_i s.t. $R_{wi} > 0$. If two senses have clusters that share a word other than w, only the cluster with the higher objective value is returned. To find the clusters displayed in Figure 3 we use this procedure with cluster size $n = 5$ on English and Russian SN vectors decomposed with basis size $k = 2000$ and sparsity $s = 4$.

B Applying WSI to Synset Matching

We use the purification procedure to represent the candidate synsets of a word w by clusters of words related to w and one if its senses a_i. First, for each synset S we define the search-space

$$V_S = \bigcup_{S' \sim S} T_{S'}$$

where $T_{S'}$ is the set of translations of lemmas of S' as in Section 3.1. Then given a word w, one of its candidate synsets S, and a fixed set size n, we run the following procedure:

1. For each sense a_i in the sparse representation (1) of w let C_i be the output cluster of the purification procedure run on sense a_i with search-space $V' = V_S$.
2. Return the (sense, sense-cluster) pair (a_S, C_S) with the highest purification procedure objective value among senses a_i.

In the following methods we will assume that each candidate synset S of w has a sense a_S and sense-cluster C_S associated to it in this way. Examples of such clusters for the French word *dalle* (flagstone, slab) are provided in Table 3.

B.1 Word-Specific Threshold

One application of Linear-WSI is the creation of a word-specific threshold α_w to use in the score-threshold method instead of a global cutoff α. We do this by using the quality of the sense cluster C_S of each candidate synset S of w as an indicator of the correctness of that synset. Recalling that u_S is the synset representation of S as a vector (see Section 3.2) and letting $f_S = f(w, a_S, C_S)$ be the objective value (2), we find α_w as follows:

1. Find $S^* = \underset{S \text{ is a candidate of } w}{\arg\max} f_S + u_S \cdot v_w$.
2. Let $\alpha_w = \min\{\alpha, u_{S^*} \cdot v_w\}$.

As this modified threshold may be lower than more than one candidate synset of w it allows for multiple synset matches even when no synset has score high enough to clear the threshold α.

B.2 Sense-Agglomeration Procedure

We use Linear-WSI more explicitly through the sense-agglomeration procedure, which attempts to recover unmatched synsets using matched synsets sharing the same sense a_i of w. We define the *cluster similarity* ρ between word-clusters $C_1, C_2 \subset V$ as the median of all cosine-similarities of pairs of words in their set product, i.e.

$$\rho(C_1, C_2) = \text{Median}\{v_x \cdot v_y : x \in C_1, y \in C_2\}.$$

Then we say that two clusters C_1, C_2 are *similar* if

$$\rho(C_1, C_2) \geq \min\{\rho(C_1, C_1), \rho(C_2, C_2)\},$$

i.e. if their cluster similarity with each other exceeds at least one's cluster similarity with itself.

Synset	Associated Sense a_S	Purified Cluster Representation C_S	Objective Value $f(w, a_S, C_S)$
flag.n.01	a_{789}	poteau (goalpost), flèche (arrow), haut (high, top), mât (matte)	0.23
flag.n.04	a_{892}	flamme (flame), fanion (pennant), guidon, signal	0.06
flag.n.06	a_{892}	dallage (paving), carrelage (tiling), pavement, pavage (paving)	0.36
flag.n.07	a_{1556}	pan (section), empennage, queue, tail	0.14
iris.n.01	a_{1556}	bœuf (beef), usine (factory), plante (plant), puant (smelly)	0.07
masthead.n.01	a_{1556}	inscription, lettre (letter), catalogue (catalog), cotation (quotation)	0.10
pin.n.08	a_{1556}	trou (hole), tertre (mound), marais (marsh), pavillon (house, pavillion)	0.17
slab.n.01	a_{892}	carrelage (tiling), carreau (tile), tuile (tile), bâtiment (building)	0.27

Table 3: Purified Synset Representations of *dalle* (flagstone, slab). Note how the correct candidate synsets (bolded) have clusters of words closely related to the correct meaning while the other clusters have many unrelated words, leading to lower objective values.

Then given a global low threshold $\beta \leq \alpha$, for each sense a_i in the sparse representation (1), sense-agglomeration consists of the following algorithm:

1. Let M_i be the set of candidate synsets S of w that have $a_S = a_i$ and score above the threshold α_w. Stop if $M_i = \emptyset$.

2. Let U_i be the set of candidate synsets S of w that have $a_S = a_i$ and score below the threshold α_w. Stop if $U_i = \emptyset$.

3. For each synset $S \in U_i$, ordered by synset-score, check that C_S is similar (in the above sense) to all clusters $C_{S'}$ for $S' \in M_i$ and has score higher than β. If both are true, add S to M_i and remove S from U_i. Otherwise stop.

The sense-agglomeration procedure allows an unmatched synset S to be returned as a correct synset of w provided it shares a sense with a different matched synset S' and satisfies cluster similarity and score-threshold constraints.