# Wordnet creation and extension made simple: A multilingual lexicon-based approach using wiki resources

Valérie Hanoka, Benoît Sagot

# Wordnet creation and extension made simple: A multilingual lexicon-based approach using wiki resources

**Valérie Hanoka**[1,2], **Benoît Sagot**[1]

1. Alpage, INRIA Paris-Rocquencourt & Université Paris 7, 175 rue du Chevaleret, 75013 Paris, France
2. Verbatim Analysis, 14 rue Friant, 75014 Paris, France
valerie.hanoka@inria.fr, benoit.sagot@inria.fr

**Abstract**

In this paper, we propose a simple methodology for building or extending wordnets using easily extractible lexical knowledge from Wiktionary and Wikipedia. This method relies on a large multilingual translation/synonym graph in many languages as well as synset-aligned wordnets. It guesses frequent and polysemous literals that are difficult to find using other methods by looking at back-translations in the graph, showing that the use of a heavily multilingual lexicon can be a way to mitigate the lack of wide coverage bilingual lexicon for wordnet creation or extension. We evaluate our approach on French by applying it for extending WOLF, a freely available French wordnet.

**Keywords:** WordNet, Word Sense Disambiguation, Wiki resources

## 1. Introduction

In recent years, researchers working on languages other than English have tried to compensate for the lack of digital lexical databases such as the Princeton WordNet (PWN) (Fellbaum, 1998). Since manual taxonomy creation is a costly investment, automatic wordnet creation and extension has become a field of interest. Many methods have been developed for creating or extending wordnets in a growing number of languages. In this paper, following previous work, we introduce a technique that leverages existing wordnets using multilingual lexical resources extracted from Wiktionaries and the Wikipedia. We apply it for extending the free French wordnet WOLF (Sagot and Fišer, 2008; Sagot and Fišer, 2012).

## 2. Related work

The question of multilinguality for bootstraping wordnets arose soon after the release of the PWN (Vossen, 1998) and still is an active research area (Pianta et al., 2002; Tufiş et al., 2004a; Diab, 2004; Sagot and Fišer, 2008; de Melo and Weikum, 2009; Navigli and Ponzetto, 2010). Multilingual approaches using the PWN as a reference for the creation of new wordnets in other languages tend to rely on the use of a similar synset inventory. Due to conceptual discrepancies between languages, finding a good translation for a given literal is not always easy.

Following works of Resnik and Yarowsky (1997) and Ide et al. (2002) on multilingual sense disambiguation, Dyvik (1998) proposed to find accurate translations of synsets literals by looking at bilingual parallel corpora. Pianta et al. (2002) used bilingual (back-)translations and contextual information (gloses) in order to ease lexicographers' work for the Italian MultiWordNet construction. Sagot and Fišer (2008) merged the results of two different approaches for synset filling: translations extracted from parallel corpora in 5 languages and bilingual lexicons extracted from freely available resources.

Beyond traditional wordnet extension/creation, some authors proposed to overcome the difficulty of constructing new wordnets for a bunch of languages other than English by using wiki resources: Navigli and Ponzetto (2010) made the link between multilingual encyclopedic knowledge from Wikipedia and multilingual wordnet creation using the PWN as a seed wordnet. de Melo and Weikum (2009) used machine learning technique over a multilingual knowledge graph (derived from mono- and bi-lingual resources) and a set of existing wordnets to build a large scale universal wordnet.

## 3. Methodology

Following the precepts of multilingual word sense disambiguation initiated by Dagan et al. (1991), we defend the idea that using a heavily multilingual lexicon is a way to mitigate the lack of wide coverage bilingual lexicon for wordnet creation or extension.

Our strategy can be sketched as follows. First, we extract a largely multilingual translation and synonym lexical database. Then, using synset-aligned multilingual wordnets, we rank translation candidates in order to favour most plausible candidates for each synset. In this section, we describe these two steps in more details. In the next sections, we report on our experiments for extending the automatically developed French wordnet WOLF using this approach, and evaluate the results.

### 3.1. Building and filtering a large-scale multilingual translation graph

The first step of our methodology consists in extracting a directed translation and synonym graph from a set of Wiktionaries and Wikipedia articles (using *inter-wiki links*) in as many languages as possible. The use of directed edges prevents pervasive effects of erroneous translations in the graph. For the sake of the explanation, the resulting graph $G$ will be referred to as the *translation graph*, synonymy links being treated as 'translations' from a language to the

same language. As this method relies on heavy multilingualism, translations for languages we did not aim to extract are kept.

For a broad group of languages, developing a simple parser for Wiktionary and Wikipedia dumps is possible. However, due to the collaborative nature of the wiki resources and the lack of a formal meta-syntax, resulting translation/synonym lists are noisy. What is more, depending on the number of languages considered for the extraction, the translation graph $G$ may quickly become very large. It is therefore necessary to filter this list.

The filtering heuristic we used is twofold.

- We remove redundant translation pairs. This step filters true duplicates and translations of various senses of the input word into a same word;
- We remove translations that involve (word, language) pairs that are not present in at least $n$ translations. In our experiments, we found that $n = 3$ offered a good trade-off between the reliability of translations and the resulting graph's coverage. This first step filters out pairs that are too isolated to be used by a multilingual approach, and allows for filtering most noise introduced by errors in the input resources. After this run, each node in the translation graph necessarily has a minimum degree of 3;

We now have built our highly-multilingual directed translation graph $G$. Each of its nodes represents a (word/term, language) pair.

### 3.2. Filling or extending synsets

As our system is designed to fill wordnet synsets using the extension approach, it requires as an input synset-aligned wordnets in $m$ different languages (ideally with $m \geq 3$). As a reminder, we shall denote by the term *synset* the union of all literals present in the synsets of all input wordnets that correspond to a same id. Consistently with this definition, we shall call *literal* a triplet of the form (word/term, language, weight). For the sake of simplicity, we shall also consider that the nodes of the translation graph $G$ represent a literal, although with a non-specified weight.

The algorithm for filling synsets we shall now present, and whose structure in pseudo-code is given in Algorithm 1, operates on a per-synset basis. It takes advantage of the directionality of the graph and simply uses the principle of back-translation in order to rank candidates.

The general idea, applied on each synset, is as follows. We start from an initial multilingual set $\Gamma_0$ filled with the literals from our input wordnets, our *gold literals* associated with a *gold weight* $\mathsf{w} = 100$. Starting from this set $\Gamma_0$, we use the translation graph $G$ to build a set $\Theta_1$ of *candidate literals*. We then put together the candidate literals from $\Theta_1$ and the literals in $\Gamma_0$, thus creating a new multilingual set $\Gamma_1$. This $\Gamma_1$ might in turn be the input of the algorithm, thus creating a new multiligual set $\Gamma_2$ via a new set $\Theta_2$ of candidate literals, and so on.

Now, let us describe our algorithm in a more formalized way. At each step, and for a given synset, we start from an input multilingual set, the set of *source literals*. These literals, of the form $(\gamma_i, l_i, \sigma_i)$, are gathered in the input multilingual set $\Gamma$. The algorithm uses $\Gamma$ to query the translation graph $G$ in order to propose a new multilingual set $\Theta$ of *candidate literals* of the form $(\theta_j, l_j, \omega_j)$. The candidate literals in $\Theta$ are then used to inflate $\Gamma$, under certain conditions.

In the reminder, we shall not explicit the difference between a node in $G$ and the literal it represents. We defined the *language degree* of a vertex $\gamma_i$ in the translation graph $G$ with regard to a language $l$, noted $deg_l(\gamma_i)$, as the number of outgoing edges in $G$ incident to $\gamma_i$ leading to literals in the language $l$. For instance, in fig.1, $deg_{tr}(table_{en}) = 2$ and $deg_{id}(table_{en}) = 1$.

**Algorithm 1** Back-Translation Algorithm

**Require:** $\Gamma_h$; $a, b \in \mathbb{N}$; $a < b$
  **for** $(\gamma_i, l_i, \sigma_i) \in \Gamma_h$ **do**
    $\Theta \leftarrow$ GETTRANSLATIONSFOR$(\gamma_i, l_i)$
    **for** $(\theta_j, l_j, \omega_j) \in \Theta$ **do**
      $\omega_{back} \leftarrow 0$
      $B \leftarrow$ GETTRANSLATIONSFOR$(\theta_j, l_j)$
      **for** $(\beta_k, l_k, \varpi_k) in B$ **do**
        **if** $\beta_k \in \Gamma_h$ **then**
          $\sigma_{\beta_k} \leftarrow$ GETSCOREIN$\Gamma_h(\beta_k)$
          **if** $\sigma_{\beta_k} \geq \mathsf{w}$ **then**
            $\omega_{back} += a$
          **else if** $\sigma_{\beta_k} \geq \mathsf{w}/5$ **then**
            $\omega_{back} += b$
          **else if** $\sigma_{\beta_k} < \mathsf{w}/50$ **then**
            $\omega_{back} -= b$
          **end if**
        **end if**
      **end for**
      **if** $\theta_j \in \Gamma_h$ **then**
        $\omega_j \leftarrow \omega_j +$GETSCOREIN$\Gamma_h(\theta_j)$
      **end if**
      $\omega_j \leftarrow \omega_j + \omega_{back}$
    **end for**
  **end for**
  $\Gamma_{h+1} \leftarrow$ MERGESETS$(\Gamma_h, \Theta)$
  **return** $\Gamma_{h+1}$

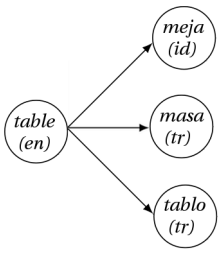


Figure 1: An illustrative snippet of a node and some of its neighbours in the translation graph

For each $\gamma_i$ that belongs to the set $S$ of *source literals*, we add in $\Theta$ all translations of $\gamma_i$ in any language found in our translation graph. Apart from its language $l$, each of these

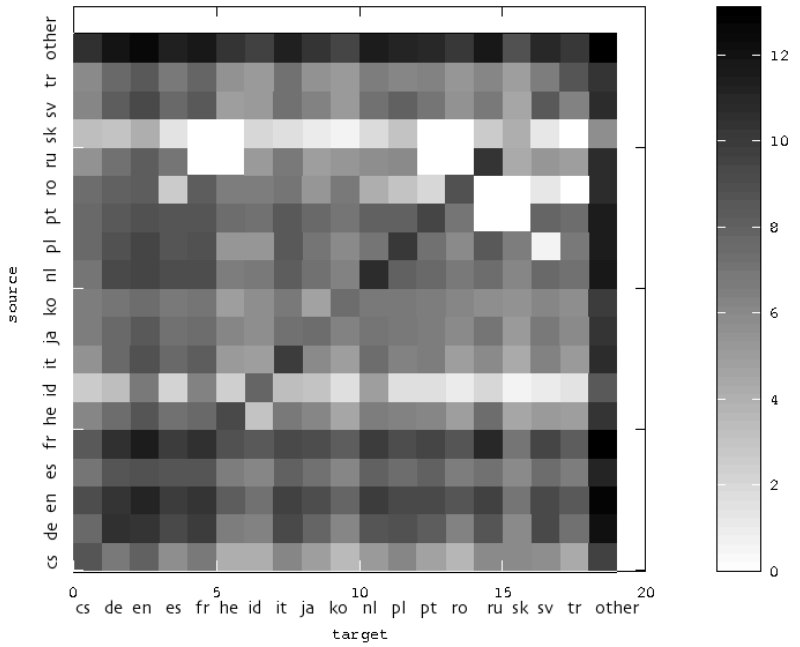


Figure 2: Log-number of translations from source to target language contained in the translation graph after filtering.

literals $\theta_i \in \Theta$ is assigned a weight $\omega_i$, initialized to the inverse of $deg_l(\gamma_i)$.

This weight is then updated heuristically according to the quality of its back-translations. Let $B$ be the set of back-translations of all $\theta_i$'s, i.e., translations of the $\theta_i$'s that belong to the set $S$ of source literals. For each $\beta_i \in B$ with weight $\varpi_i$:

- if $\varpi_i \geq \mathsf{w}$, we add a back-translation bonus $a$ to $\omega_i$;
- if $\varpi_i \geq \frac{\mathsf{w}}{5}$, we add a reduced back-translation bonus $b$ to $\omega_i$;
- if $\varpi_i < \frac{\mathsf{w}}{10}$, we subtract the reduced back-translation bonus $b$ to $\omega_i$;

Different bounds and bonus values can be tried. For our experiment, we choose empirically $a = \frac{\mathsf{w}}{10}$ and $b = \frac{\mathsf{w}}{50}$

Once the set $\Theta$ of candidate literals is produced, we create a new multilingual set $\Gamma'$ by taking the union of the input multilingual set $\Gamma$ and the content of $\Theta$.

As explained above, this algorithm might be repeated several times, taking the output multilingual set of one step as the input multilingual set of the next one. In our experiments, we executed it twice.

After having applied this algorithm twice on each synset, we enriched all synsets in our input worndets and produced an expanded wordnet in which each (literal, synset) pair has retained its final output weight. This allowed for filtering out new (literal, synset) pairs that have a score lower than a certain threshold.

Before we illustrate this algorithm on an example, we now provide details about our experiments.

## 4. Experiments

We extracted translation/synonym pairs from a set of Wiktionaries in 18 languages (Czech, Dutch, English, French, German, Hebrew, Indonesian, Italian, Japanese, Korean Polish, Portuguese, Romanian, Russian, Slovak, Spanish, Swedish and Turkish) and the French Wikipedia. The first raw extraction step resulted in 9,916,345 translations pairs and 748,385 synonymy pairs. After filtering 1,584,370 ($16\%$) translations pairs and 366,826($49\%$) synonymy pairs remained. The average degree (both directions) of the translation graph is $3.522$.

Figure 2 shows the number of translations pairs before and after filtering for each language.

For our experiment, we used the PWN 2.0 for English, and the BalkaNet wordnets for Czech, Bulgarian and Romanian (Tufiş et al., 2004b) which are aligned to the PWN 2.0. However, we did not use the French WOLF, in order to compare our approach to those used for developing this resource. Here is an example of the algorithm applied to synset ENG20-01562015-a {obedient}. Table 1 presents the aligned synset after completion.

For this example, we have $\Gamma = \{$(inequality, en, 100), (imparitate, ro, 100), (inegalitate, ro, 100), (неравенство, bg, 100)$\}$ for the first run.

Translations for *inequality(en)* result in 7 candidates literals in 6 different languages (de, it, fr, sv, en, es). Among them, three candidate literals gain the back-translation bonus $a$: *Ungleichung(de)*, *inégalité(fr)*, *olikhet (sv)*.

No translations were found for *imparitate(ro), inegalitate(ro) and неравенство(bg)*.

| English | Romanian | Bulgarian | French |
|---|---|---|---|
| **inequality**<br>triangle inequality(2.0)<br>dissimilarity(1.5) | **imparitate**<br>**inegalitate** | **НЕРАВЕНСТВО** | <br>inégalité (12.0)<br>dissemblance (1.0)<br>inéquation (1.0) |

Table 1: Synset ENG20-04543367-n aligned in different languages after completion. Bold words represents gold literals. Candidate literals are followed by their score. Empty column for Czech is omitted here.

The preceding step results in a bigger $\Gamma$ set whose cardinal is now $|\Gamma| = 11$, as 7 candidates literals have been added. For the second run of the algorithm:

Translations for *inequality(en)* result in updating the scores of the 7 candidate literals added during last run, adding bonus to the same trio as before.

Translations for new item *Ungleichheit(de)* whose score is 0.8 result in updating scores for 1 literal. Four new candidate literals are added in 4 languages (tr, fr, en, ru), among which one gain the $a$ bonus (*eşitsizlik(tr)*).

Translations for new item *Ungleichung(de)* whose score is 11.0 result in updating scores for 1 literal. Two new candidate literals are added in 2 languages (fr, ru), among which one gain the $a$ bonus (*inéquation(fr)*).

Translations for new item *inégalité(fr)* whose score is 12.0 result in updating scores for 2 English literals.

Translations for new item *olikhet(sv)* whose score is 12.0 result in updating scores for 1 literal, and add a new candidate literal in Swedish with no bonus.

No translations were found for НЕРАВЕНСТВО*(bg)* and the other 5 candidate literals added during the first run.

The final result, as showed in table 1, managed to guess related *candidate literals* for French and English. The most accurate French candidate (*inégalité*) receive by far the best score among other French candidates (*inéquation, dissemblance*). Concerning the English candidates, poor scores show that they did not gain any bonus during the process, and should probably not be accepted in the synset.

## 5. Results and evaluation

Retaining only candidates with a score greater than 30, we created $10,568$ (literal, synset) candidates, among which as many as $6,119$ ($58\%$) are not included in the WOLF. Table 2 show a random sample of these (literal, synset) candidates, associated with PWN literals and definition for the synset, the score of the candidate, our manual evaluation (YES if it is correct, NO otherwise), as well as information about whether this candidates was already included in the WOLF (YES if it was in the WOLF, NO otherwise).
For evaluating our approach, we put together all (literal, synset) pairs we produced as well as all WOLF (literal, synset) pairs for synsets for which we generated at least one candidate. We performed a manual evaluation of $400$ of these (literal, synset) pairs by assigning a boolean score (correct/incorrect) to all such pairs for a random sample of the synsets.

We evaluated our approach according to two parameters. The first one, $t$, is a threshold on the score associated by our approach to each candidate: setting the threshold at 30 retains all candidates, whereas setting it at a higher value discards all candidates with a lower score. The second parameter $n_{\max}$ is an upper bound on the number of candidates retained for each synset: if this parameter is set to 3, the candidates for a given synset are sorted by decreasing score, and at most the first 3 candidates are retained (less than 3 if scores fall below $t$).
For each value of $t$ and each value of $n_{\max}$ we evaluate:

- the **precision** of our candidates, the ratio of correct candidates w.r.t. the total number of candidates; if we retain all $10,568$ candidates (i.e., $t = 30$ and $n_{\max} = \infty$), the precision is $74.1\%$;
- an estimation of the **number of correct candidates** we obtain, computed as the total number of candidates times the precision figure; again, if we retain all candidates, we can thus expect around $10,568 \times 0.741 = 6,465$ candidates; among them, $6,119$ are not in the WOLF, and their precision is $65\%$; this shown that not only our approach generated candidates not generated by previous approaches, but these candidates have a high precision as well; still, using more strict parameters can improve this precision figure (see below);
- an **approximate "recall"**, measured w.r.t. these $6,465$ correct candidates that are kept if we retain them all (i.e., with $t = 30$ and $n_{\max} = \infty$);
- an **approximate "f-score"** based on the precision and "recall" figures.

Table 3 shows the precision figures and associated number of candidates that we obtain with different sets of parameter values. The outcome of these evaluations can be summarized as follows. First, as expected, increasing $n_{\max}$ or $t$ decreases the number of candidates but increases the precision. For $n_{\max} = 1$ and $t = 50$, the precision reaches $86\%$ over $3,353$ candidates. Among them, $1,601$ candidates are not present in the WOLF, and the precision on these candidates is as high as $82\%$. On candidates already in the WOLF, the precision is over $89\%$, vs. $87\%$ on all (literal, synset) pairs from the WOLF. However, such parameter values are quite restrictive. Based on the "f-score" figures, the optimal parameter values are $(t = 30, n_{\max} = 3)$ With such values, we retain $10,403$ candidates (almost all of them) with a $74,8\%$ precision (slightly but significantly higher than with $n_{\max} = \infty$. A manual validation of all these candidates would add around $7,781$ correct candidates in the WOLF.

| French literal | Synset id | Score | PWN literals in the synset | PWN definition | Correct (man. eval.) | Already in WOLF |
|---|---|---|---|---|---|---|
| harmonie | 06738523-n | 54.7 | harmony, concord, concordance | agreement of opinions | YES | NO |
| ivre | 00879266-a | 100.0 | intoxicated, drunk | as if under the influence of alcohol | YES | NO |
| alphabet | 06096415-n | 39.6 | alphabet | a character set that includes letters and is used to write a language | YES | NO |
| ensemble | 00515754-b | 66.3 | together | in each other's company | YES | NO |
| jeunesse | 10099908-n | 47.0 | young person, youth younker, spring chicken | a young person (especially a young man or boy) | NO | YES |
| tête | 08134688-n | 100.0 | head | the top of something | YES | YES |
| salamandre | 03825556-n | 35.6 | poker, stove poker, fire hook, salamander | fire iron consisting of a metal rod with a handle; used to stir a fire | NO | NO |
| périlleux | 01991204-a | 40.6 | hazardous, risky, venturesome, venturous | involving risk or danger | YES | YES |
| accord | 06733497-n | 34.7 | agreement | the verbal act of agreeing | YES | YES |
| électricité | 07054143-n | 71.8 | electricity | keen and shared excitement | NO | YES |

Table 2: Examples of (literal, synset) candidates produced by our algorithm

| | $t = 30$ | $t = 40$ | $t = 50$ | $t = 60$ |
|---|---|---|---|---|
| $n_{\max} = 1$ | 8362/77.3 | 5340/81.5 | 3353/85.6 | 2245/90.5 |
| $n_{\max} = 3$ | 10403/74.8 | 6298/80.6 | 3890/85.1 | 2582/89.6 |
| $n_{\max} = \infty$ | 10568/74.1 | 6357/80.3 | 3917/85.2 | 2594/89.6 |

Table 3: Number of (literal, synset) candidates retained when using different values for the thresholds $t$ and $n_{\max}$, with the corresdonding precision measure.

As can be seen on Table 3, precision figures consistently correlate with $t$ and with $n_{\max}$, which shows the relevance of the score computed by our algorithm. As a consequence, higher $t$ values associated with $n_{\max} = 1$ lead to higher precision figures. For example, parameter values $(t = 60, n_{\max} = 1)$ lead to as high a precision as $90,5\%$, still retaining $2,245$ candidates among which around $2,031\%$ correct ones.

Chosing $(t = 30, n_{\max} = 3)$ as parameters, we performed a manual examination of correct generated candidates that are not in the WOLF. In almost all cases, these candidates correspond to high or medium frequency words that are polysemic, sometimes highly polysemic. In other words, these candidates correspond to the most useful information in any wordnet-based application, but also to cases for which previous approaches used for developing the WOLF performed the least satisfyingly: the alignment-based approach used for disambiguating polysemous words (Sagot and Fišer, 2008) was highly corpus-dependant and could not cover well medium frequency words; the lexicon-based approach was first restricted to monosemous words (Sagot and Fišer, 2008), and later efforts towards its use on polysemous words (Sagot and Fišer, 2012) did not perform well on highly polysemous words. Examples of literals not present at all in WOLF but covered by our candidates include as basic and frequent words as *manger* 'eat', *taper* 'hit', *lent* 'slow', *faim* 'hunger' or *dehors* outside'.

## 6. Conclusion and perspectives

The method presented in this paper consists in extending/-bootstraping synset-aligned wordnets simply by looking at back-translations found in a large multilingual translation graph extracted from a set of wiki resources in as many languages as possible. It is well suited for creating or enriching WordNets for languages that have at their disposal large or medium coverage Wiktionary and Wikipedia. An interesting point would be to determine if, for languages with smaller wiktionary and/or wikipedia, the use of larger wiktionaries for other languages and/or the improvement of translation graph can prove to be sufficient for a good quality extension. We plan to improve current results by upgrading the translation graph's quality as in Mausam et al. (2009) and computing scores for candidate literals by doing a march in the translation graph.

## 7. References


Ido Dagan, Alon Itai, and Ulrike Schwall. 1991. Two languages are more informative than one. In *in Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 130–137.

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*, CIKM '09, pages 513–522, New York, NY, USA. ACM.

Mona T. Diab. 2004. The feasibility of bootstrapping an arabic wordnet leveraging parallel corpora and and an english wordnet. In *Proceedings of the Arabic Language Technologies and Resources*, Cairo.

Helge Dyvik. 1998. Translations as semantic mirrors: from parallel corpus to wordnet. In *Proceedings of the Workshop Multilinguality in the lexicon II at the 13th biennial European Conference on Artificial Intelligence (ECAI'98)*, pages 24–44, Brighton, UK.

Christiane Fellbaum, editor. 1998. *WordNet: An Elec-*

*tronic Lexical Database*. MIT Press, Cambridge, Massachusetts.

Nancy Ide, Tomaž Erjavec, and Dan Tufiş. 2002. Sense discrimination with parallel corpora. In *Proceedings of the ACL-02 workshop on Word sense disambiguation: recent successes and future directions - Volume 8*, WSD '02, pages 61–66, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mausam, Stephen Soderland, Oren Etzioni, Daniel S. Weld, Michael Skinner, and Jeff Bilmes. 2009. Compiling a massive, multilingual dictionary via probabilistic inference. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 262–270, Stroudsburg, PA, USA. Association for Computational Linguistics.

Roberto Navigli and Simone Paolo Ponzetto. 2010. BabelNet: Building a Very Large Multilingual Semantic Network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 216–225, Uppsala, Sweden.

Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. 2002. Multiwordnet: developing an aligned multilingual database. In *Proceedings of the First International Conference on Global WordNet*.

Philip Resnik and David Yarowsky. 1997. A perspective on word sense disambiguation methods and their evaluation. In *Proceedings of the SIGLEX Workshop*.

Benoît Sagot and Darja Fišer. 2008. Building a free french wordnet from multilingual resources. In *Ontolex 2008*, Marrakech, Morocco.

Benoît Sagot and Darja Fišer. 2012. Automatic extension of WOLF. In *Proceedings of the 6th Global WordNet Conference*, Matsue, Japan.

D. Tufiş, D. Cristea, and S. Stamou. 2004a. Balkanet: Aims, methods, results and perspectives. a general overview. *In: D. Tufiş (ed): Special Issue on BalkaNet. Romanian Journal on Science and Technology of Information*, pages 3–4.

Dan Tufiş, Dan Cristea, and Sofia Stamou. 2004b. BalkaNet: Aims, Methods, Results and Perspectives. A General Overview. In *Romanian Journal on Information Science and Technology. Special Issue on BalkaNet*, volume 7, pages 9–34.

Piek Vossen, editor. 1998. *EuroWordNet: a multilingual database with lexical semantic networks*. Kluwer Academic Publishers, Norwell, MA, USA.