Automatic Evaluation of Quality of an Explanatory Dictionary by Comparison of Word Senses*

Alexander Gelbukh[1,2], Grigori Sidorov[1],
SangYong Han[2], and Liliana Chanona-Hernandez[3]

[1] Center for Computing Research (CIC), National Polytechnic Institute (IPN),
Av. Juan de Dios Bátiz, Zacatenco, CP 07738, Mexico City, Mexico
{gelbukh,sidorov}@cic.ipn.mx; www.gelbukh.com

[2] Department of Computer Science and Engineering, Chung-Ang University,
221 Huksuk-Dong, DongJak-Ku, Seoul, 156-756, Korea
hansy@cau.ac.kr

[3] Department of Computer Science, Las Americas University,
Mexico City, Mexico
lchanona@mail.com

Abstract. Words in the explanatory dictionary have different meanings (senses) described using natural language definitions. If the definitions of two senses of the same word are too similar, it is difficult to grasp the difference and thus it is difficult to judge which of the two senses is intended in a particular contexts, especially when such a decision is to be made automatically as in the task of automatic word sense disambiguation. We suggest a method of formal evaluation of this aspect of quality of an explanatory dictionary by calculating the similarity of different senses of the same word. We calculate the similarity between two given senses as the relative number of equal or synonymous words in their definitions. In addition to the general assessment of the dictionary, the individual suspicious definitions are reported for possible improvement. In our experiments we used the Anaya explanatory dictionary of Spanish. Our experiments show that there are about 10% of substantially similar definitions in this dictionary, which indicates rather low quality.

1 Introduction

Words in a typical explanatory dictionary have different senses, while only one of which can be meant for a particular occurrence of the word in a particular text. For example, the word "bank" can stand for a financial institution, shore,

* Work done under partial support of Mexican Government (CONACyT, SNI), IPN (CGEPI, COFAA, PIFI), and Korean Government (KIPA Professorship for Visiting Faculty Positions in Korea). The first author is currently on Sabbatical leave at Chung-Ang University.

set, etc., depending on the context. To determine the particular word sense in a given text is important for a number of purposes, such as:

- For any kind of understanding of the text. For example, if you are advised to keep your money in a reliable *bank*, would you carry it to a financial institution or a river shore?
- For translating the text. For example, "*my account in the* bank" is to be translated into Spanish as "*mi cuenta en el* banco", whereas "*on the* bank *of the lake*" as "*en la* orilla *del lago*."
- For information retrieval. For example, for a query "*branch banks in Chicago*", a document containing "*There are two Citybank branch banks in the central area of Chicago*" is relevant while "*The hotel is under maple branches at the beautiful bank of Michigan Lake, with an amazing view of Chicago's skyscrapers*" is not.

Such tasks can be performed either manually or automatically. In case of automatic analysis, the task of selection of the intended sense in a given context is called (automatic) word sense disambiguation (WSD) task. This task proved to be quite difficult and is nowadays the topic of active research.

The senses are defined in explicative dictionaries by means of definitions, the latter being the only source of information on a particular sense. A typical WSD algorithm tries to compare the definition of the sense with the surrounding words to guess the meaning intended in a particular context [4]. Thus the quality of disambiguation greatly depends on two factors related to the dictionary that defines the senses:

- Which cases of the use of the given words are considered different senses,
- How these senses are described in the dictionary.

Many dictionaries, including WordNet [11], have been extensively criticized for too fine-grained sense distinction that makes it difficult to choose a particular sense out of a set of very similar or closely related senses. In addition, the difference between very similar senses can disappear in most cases. For example, suppose for the word *window*, a dictionary distinguishes between:

1. the interior of the hole in the wall ("*look through the window*"),
2. the place in the wall where the hole is ("*near the window*"),
3. a frame with the glass in the hole is ("*break the window*"),
4. a computer program interface object ("*the button in the main window*").

Then in many cases can say for sure that in a particular text (for example, "*John came up to the window, broke it and fell out of it*") one of the first three senses was meant (and not the fourth one) but will have difficulties guessing which one of the three because they are too similar. This can happen either because of too fine-grained sense distinction or because of poor wording of the definitions.

Automatic WSD programs are especially sensible to this problem. In frame of WSD it has been even suggested that for practical purposes the senses in the dictionaries should be clustered together to form coarser classifications easier to make clear distinctions [5, 7].

In this paper we suggest a method for automatic detection of this problem. It has at least two potential uses:

- To help the lexicographer compiling or maintaining the dictionary to detect potential problems in the system of senses leading to error-prone interpretation,
- To help a human designer or even a software agent [1] to choose a better dictionary to use out of a set of available dictionaries.

Namely, we suggest that in a "good" (for the purposes involving WSD) dictionary the senses of each word are described differently enough to be clearly distinguishable. On the other hand, too similar descriptions of senses of some word indicate a potential problem with the use of the dictionary. To evaluate the similarity between different senses of a word we calculated the share of "equal" words in their definitions in the given explanatory dictionary.

The notion of "equal words" needs clarification. Obviously, equal letter strings represent the same word. In addition, we consider different morphological forms ("*take*" = "*takes*" = "*took*"...) to represent the same word; the process of their identification is called stemming [8]. Finally, we consider two words to be equal if they are synonymous.

We implemented our method for the Anaya explanatory dictionary of Spanish language. We used a synonym dictionary of Spanish for detecting synonyms.

In the rest of the paper, we discuss the algorithm we use for finding similar definitions, the present and the experimental results, and finally draw some conclusions.

2 Algorithm

We calculate a numerical value characterizing the quality of a dictionary as the average similarity between two different word senses of the same word. We also detect specific senses that are too similar, for possible manual merging or improvement of their definitions.

To measure the similarity between two texts (two definitions, in our case) we use an expression based on the well-known Dice coefficient [3, 9]. The latter is defined as follows:

$$D(t_1, t_2) = \frac{2 \times W_1 \cap W_2}{|W_1| + |W_2|}, \tag{1}$$

where W_1 and W_2 are sets of words in texts t_1 and t_2. This expression characterizes how many words the texts have in common if the words are compared literally. $W_1 \cap W_2$ means that we take the words that exist in both texts.

To take synonymy into account we use a similar but slightly modified expression. Let $W_1 \circ W_2$ be the set of synonymous words in the two definitions calculated as described in 1. Then we determine the similarity as follows:

$$D(t_1, t_2) = \frac{|W_1 \cap W_2| + k\,|W_1 \circ W_2|}{\max(|W_1|, |W_2|)}, \tag{2}$$

pre-process all dictionary: POS-tagging, stemming
for each dictionary entry
  for each homonym of the word
    for each unordered pair of its senses $S_1 \neq S_2$
      D = $\frac{similarity(S_1,S_2)}{\max(length(S_1),length(S_2)}$ by (2)
estimate the average D and report too high values of D

Function $similarity(S_1, S_2)$:
  $similarity = 0$; $matched\_words\_list = \emptyset$
  for each word $w_i \in S_1$ except stop-words
    if $w_i \in S_2$ then
      add 1 to $similarity$; add w_i to $matched\_words\_list$
    else for each synonym $s \in synonyms(w_i)$
      if $s \notin matched\_words\_list$ and $s \in S_2$ then
        add k to $similarity$; add s to $matched\_words\_list$ (we use $k = 1$)
        break the cycle
  for each word $w_j \in S_2$ except stop-words
    for each synonym $s \in synonyms(w_j)$
      if $s \notin matched\_words\_list$ and $s \in S_1$ then
        add k to $similarity$; add s to $matched\_words\_list$
        break the cycle

Fig. 1. The algorithm

We use the maximum value for normalization, because all words from the longer definition can be synonymous to the words from the other one. Consequently we do not multiply the expression by 2 as in (1). The weighting coefficient k reflects the degree of importance of the synonyms. We use the value $k = 1$ since for the purposes of our experiment it is important to measure the maximum possible similarity. Besides, in our opinion, synonyms should be treated in this calculation with the same importance as literal word intersection, because, by definition, synonyms have similar meanings and normally distinguish only the shades of meaning. Thus, though it is not always possible to substitute one synonym with another in the text, they do express more or less the same concept.

Our algorithm is shown in 1. For each word in the dictionary, we measured the similarity between its senses. Obviously, words with only one sense were ignored. Since similarity is a symmetrical relation, it was calculated only once for each pair of senses of the word.

Note that we considered homonyms as different words and not as different senses. Normally, this is the way that they are represented in the dictionaries—as different groups of senses. Besides, the homonyms usually have different meanings, so they are of little interest for our investigation.

We measure the similarity in the following way. At the beginning, the similarity score is zero. The words are taken one by one from the first sense in the pair and they are searched in the other sense. If the word is found then the score is incremented. If the word is matched, it is marked (added to the list of matched

synonyms) to avoid counting it in the opposite direction while processing the other sense in the pair.

Note that we use normalized words with POS-tags, which allows for matching of any grammatical form. Additionally, this allows for taking into account only significant words and discarding auxiliary words. Unknown words are processed as significant words but only if their length is greater than a given threshold. We used the threshold equal to two letters, i.e., we considered all too short words to be stop-words. We also ignored auxiliary words because normally they do not add any semantic information (like conjunctions or articles), or add some arbitrary information (like prepositions, which frequently depend on the subcategorization properties of verbs).

If the word has no match in the other sense, we search its synonyms in the dictionary of synonyms and check them one by one in the other sense. If the synonym is matched, the score is incremented. At the same time the synonym is marked (added to an auxiliary list of matched synonyms) to avoid repetitive counting. The already marked synonyms are not used for comparisons.

At the next step, the procedure is repeated for the other sense in the pair but only matching of synonyms is performed because the literally matching words have been found already. At the same time, synonyms yet can give additional score because we cannot guarantee that our synonym dictionary is symmetrical (i.e., if A is synonym for B, then B is synonym for A).

Finally, we apply the formula 2 for calculating the modified coefficient of similarity.

3 Experimental Results

We used Anaya dictionary as a source of words and senses. This dictionary has more than 30,000 headwords that have in total more that 60,000 word senses; only 17,000 headwords have more than one sense. We preferred this dictionary to Spanish WordNet (see, for example, [11]) because Spanish WordNet has definitions in English while we were interested in Spanish and used the linguistic tools and synonym dictionary for Spanish.

For morphological processing, we applied Spanish morphological analyzer and generator developed in our laboratory [2].

All definitions in the dictionary were normalized and the POS-tagging procedure was applied [10]. In the experiment, we ignore the possible word senses that were assigned to each word in definition (see [10]) because this procedure gave a certain percent of errors that we try to eliminate here. Word senses in definitions can be taken into account in future experiments.

We also used a synonym dictionary of Spanish that contains about 20,000 headwords. This dictionary is applied at the stage of measuring of the similarity between senses for detecting synonymous words in definitions.

Below we give an example of the data (normalized definitions from Anaya dictionary) that was used. The definition of the word *abad* in one of the senses is as follows:

Abad = Título que recibe el superior de un monasterio o el de algunas colegiatas. (*Abbot = Title that receives a superior of a convent or of some churches*).

The normalized version of this definition is as follows:

Abad = título#noun que#conj recibir#verb el#art superior#noun de#prep un#art monasterio#noun o#conj el#art de#prep alguno#adj colegiata#noun .#punct

(*Abbot = title#noun that#conj receive#verb a#art superior#noun of#prep a#art convent#noun or#conj of#prep some#adj church#noun*),

where *#conj* is conjunction, *#art* is article, *#prep* is preposition, *#adj* is adjective, *#punct* is punctuation mark, and *#noun* and *#verb* stay for noun and verb. There are some words that were not recognized by our morphological analyzer (about 3%), which are marked as *#unknown*.

The results of applications of the algorithm shown in Figure 1 are presented in Table 1. Since the similarity is a fraction, some fractions are more probable than others; specifically, due to a high number of short definitions, there were many figures with small denominators, such as $\frac{1}{2}$, $\frac{1}{3}$, $\frac{2}{3}$, etc. To smooth this effect, we represented the experimental results by intervals of values and not by specific values. We present the results for zero similarity separately because there were many senses without any intersection. As one can see, about 1% of

Table 1. Number of sense pairs per intervals.

Interval of similarity	Number of sense pairs	Percent of sense pairs
0.00–0.01	46205	67.43
0.01–0.25	14725	21.49
0.25–0.50	6655	9.71
0.50–0.75	600	0.88
0.75–1.00	336	0.49

sense pairs are very similar—they contain more than 50% of equivalent words (with synonyms), and about 10% of sense pairs are significantly similar—they contain more than 25% of equivalent words. 10% is rather large value, so these definitions should be revised in the Anaya dictionary to improve its quality (at least for the purposes of WSD.

4 Conclusions

We proposed a method of automatic evaluation of the quality of the explanatory dictionary by comparison of different senses of each word. We detect and report to the developer the pairs of senses with too similar definitions. We also assess

the total quality of the dictionary (with respect to the clarity of distinction of the senses) as an inverse value to the average similarity of the senses of each word. Though it is only one aspect of quality of the dictionaries, it is a rather important feature for all tasks involving WSD, both automatic (for a computer) and manual (for a human user of the dictionary).

The method consists in calculating the relative intersection of senses considered as bags of words; the words are previously POS-tagged and stemmed. During this comparison, both literal intersection and intersection of synonyms of the words are taken into account.

The experiment was conducted for the Anaya dictionary of Spanish. The results show that about 10% of sense pairs are substantially similar (more than 25% of the similar words). This is rather large percentage of similar senses, thus, the dictionary can be improved by rewprding of these definitions, merging these senses, or restructuring the whole entry.

In the future, we plan to perform such experiments with other dictionaries, like, for example, WordNet, and with other languages.

References

1. A. Burton-Jones, V. C. Storey, V. Sugumaran, P. Ahluwalia. Assessing the Effectiveness of the DAML Ontologies for the Semantic Web. In: *Proc. NLDB-2003, Applications of Natural Language to Information Systems*. Lecture Notes in Informatics, Berlin, 2003.
2. Gelbukh, A. and G. Sidorov (2002). Morphological Analysis of Inflective Languages Through Generation. *J. Procesamiento de Lenguaje Natural*, No 29, September 2002, Spain, pp. 105–112.
3. Jiang, J.J. and D.W. Conrad. (1999) From object comparison to semantic similarity. In: *Proc. of Pacling-99 (Pacific association for computational linguistics)*, August, 1999, Waterloo, Canada, pp.256–263.
4. M. Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone. In *Proceedings of SIGDOC '86*, 1986.
5. Magnini, B., Strapparava, C. Experiments in Word Domain Disambiguation for Parallel Texts. In: *Proceedings of the ACL Workshop on Word Senses and Multilinguality*, Hong Kong, China, 2000.
6. Manning, C. D. and Shutze, H. (1999) *Foundations of statistical natural language processing.* Cambridge, MA, The MIT press, 680 p.
7. A. Suárez and M. Palomar. Word Sense vs. Word Domain Disambiguation: a Maximum Entropy approach In: *Proc. TSD-2002, Text, Speech, Dialogue*. Lecture Notes in Artificial Intelligence, Spriner-Verlag, 2002.
8. Porter, M.F. An algorithm for suffix stripping. *Program*, **14**(3):130-137, 1980.
9. Rasmussen E. (1992) Clustering algorithms. In: Frakes, W. B. and Baeza-Yates, R. *Information Retrieval: Data Structures and Algorithms.* Prentice Hall, Upper Saddle River, NJ, 1992, pp. 419–442.
10. Sidorov G. and A. Gelbukh (2001). Word sense disambiguation in a Spanish explanatory dictionary. In: *Proc. of TALN-2001, Traitement Automatique du Langage Naturel*. Tours, France, July 2-5, 2001, pp 398–402.
11. *WordNet: an electronic lexical database.* (1998), C. Fellbaum (ed.), MIT, 423 p.