Graph Based Algorithm for Automatic Domain Segmentation of WordNet

Brijesh Bhatt   Subhash Kunnath   Pushpak Bhattacharyya
Center for Indian Language Technology
Indian Institute of Technology Bombay
Mumbai, India
{ brijesh, subhash, pb } @cse.iitb.ac.in

Abstract

We present a graph based algorithm for automatic domain segmentation of Wordnet. We pose the problem as a Markov Random Field Classification problem and show how existing graph based algorithms for Image Processing can be used to solve the problem. Our approach is unsupervised and can be easily adopted for any language. We conduct our experiments for two domains, health and tourism. We achieve F-Score more than .70 in both domains. This work can be useful for many critical problems like *word sense disambiguation, domain specific ontology extraction etc.*

1 Introduction

Over the years, Wordnet has served as an important lexical resource for many Natural Language Processing (NLP) applications. Picking up a right sense of a word from the fine grained sense repository of Wordnet is at the heart of many NLP problems. Many researchers have used Wordnet for domain specific applications like *word sense disambiguation* (Magnini et al., 2002a; Khapra et al., 2010), *domain specific taxonomy/ontology extraction* (Cimiano and Vlker, 2005; Yanna and Zili, 2009) etc. These applications rely on 'One sense per discourse' (Gale et al., 1992) hypothesis to identify domain specific sense of a word. 'Dividing Wordnet's lexical and conceptual space into various domain specific subspace can significantly reduce search space and thus help many domain specific applications' (Xiaojuan and Fellbaum, 2012).

With the purpose of categorizing Wordnet senses for different domain specific applications, Magnini and Cavagli (2000) constructed a domain hierarchy of 164 domain labels and annotated Wordnet synsets with one or more label from the hierarchy. The categories were further refined by linking domain labels to subject codes of Dewey Decimal Classification system (Bentivogli et al., 2004). Beginning with Wordnet 2.0, Domain category pointers were introduced to link domain specific synsets across part of speech. However, the manual determination of a set of domain labels and assigning them to Wordnet synsets is a time consuming task. Also, the senses of words evolves over a period of time and accordingly Wordnet synsets also undergo changes. This makes the static assignment of domain label a costly exercise.

With the intention to reduce manual labor of domain categorization and to facilitate use of Wordnet in domain specific applications, there has been efforts to (semi) automatically assign domain labels to Wordnet synsets. Most of these efforts rely on Wordnet concept hierarchy and use label propagation schemes to propagate domain labels through the hierarchy. However, the heterogeneous level of generality poses a key challenge to such approaches. For example, 'Under Animal (subsumed by Life_Form) we find out specific concepts, such as Work_Animal, Domestic_Animal, kept together with general classes such as Chordate, Fictional_Animal, etc.'(Gangemi et al., 2003). Another key challenge in assigning the domain labels is the quality of domain hierarchy and semantic distance between domain labels (Xiaojuan and Fellbaum, 2012).

In this paper, we present a corpus based approach for automatic domain segmentation of Wordnet. The aim of our work is to provide a general solution that can be used across languages to construct domain specific conceptualization from Wordnet. The proposed system works in two steps,

1. We construct domain specific conceptualiza-

tion from the corpus.

2. The domain specific conceptualization is then disambiguated and linked to Wordnet synsets to generate domain labels.

We pose Wordnet domain segmentation as an image labeling problem and use existing techniques in the field of image processing system to solve Wordnet domain labeling problem. The proposed method is completely unsupervised and requires only Part Of Speech tagged corpus. Hence, it can be easily adopted across languages. Our method also does not require any predefined set of domain category labels, however if such labels are available it can be incorporated into system to generate better labeling.

The remaining of the paper is organized as follows, section 2 describes related work. Section 3 describes the proposed graph based algorithm for Wordnet domain labeling. Section 4 and 5 discuss the experiments and conclusion.

2 Related Work

Two major attempts to categorize Wordnet synsets are Wordnet Domain (Magnini and Cavagli, 2000) and Wordnet Domain Category pointers. In this section we first present a brief overview of these efforts and then describe some efforts to automate the task of domain labeling of Wordnet synsets. We also mention the attempts made for other languages apart from English.

2.1 Wordnet Domain Hierarchy and Domain Category Pointers

Domain categorization of Wordnet synset has been an active area of research for more than a decade now. Magnini and Cavagli (2000) have developed Wordnet Domain Hierarchy (WDH) by annotating Wordnet1.6 using 250 Subject Field Codes (SFC). They used semi-automated approach in which the top level concepts are manually marked with SFC and then the labels are automatically propagated through the hierarchy. Finally, the labeling is again evaluated and refined manually. The semantic structure of WDH was further refined by Bentivogli et al. (2004).

Starting from Wordnet 2.0, *domain category pointers* were introduced in the Wordnet. 'Unlike the original Wordnet Domain, the domain category pointers use Wordnet synsets as domain labels and synsets across part of speech are linked through domain pointers' (Xiaojuan and Fellbaum, 2012). However, 'only 5% of Wordnet 3.0 synsets are linked to 438 domain categories and out of these linked synsets only 30% synsets have same label in both Wordnet Domain and Domain Category'.

2.2 Automated Approaches

Considering the growing size of Wordnet and the amount of efforts required to construct domain categories, it is apparent to develop semi-automated or automated methods for domain categorization of Wordnets. One of the earlier efforts in this direction was by Buitelaar and Sacaleanu (2001). They extracted domain specific terms using tf*idf measure and then disambiguated these terms using GermaNet synsets. The disambiguation was performed based on the assumption that the hypernymy and hyponymy terms are more likely to have same domain label. Magnini et al. (2002b) have performed a comparative study of corpus based and ontology based domain annotation. They have used frequency of words in the synonym set as a measure to identify domain of a synset.

Gonzalez-Agirre et al. (2012) have proposed a semi-automatic method to align the original Wordnet 1.6 based domains to Wordnet 3.0. They have used domain labels already assigned to some top level synsets and then propagated the domain label across Wordnet hierarchy using UKB algorithm (Agirre and Soroa, 2009). Their approach is based on an assumption that 'A synset directly related to several synsets labeled with a particular domain (i.e biology) would itself possibly be also related somehow to that domain (i.e. biology)'(Gonzalez-Agirre et al., 2012).

Fukumoto and Suzuki (2011) have adopted a corpus based approach to assign domain labels to Wordnet synsets. They first disambiguate the corpus words with Wordnet senses and then use Markov Random Walk based Page Rank Algorithm to rank domain relevance of Wordnet senses. Zhu et al. (2011) have proposed gloss based disambiguation technique for domain assignment to Wordnet synset. They used existing domain labels of Wordnet 3.0 and predicted domains based on words in the gloss of the synsets.

There have also been efforts to adopt English Wordnet domain labels for other languages. Lee et al. (2009) have used English-Chinese Wordnet mapping to domain tag Chinese Wordnet.

2.3 Proposed Approach

Like Buitelaar and Sacaleanu (2001), Magnini et al. (2002b) and Fukumoto and Suzuki (2011), we also follow corpus based approach for Wordnet Domain Labeling. Key points of difference among these approaches can be summerized as follows,

1. Both Buitelaar and Sacaleanu (2001) and Magnini et al. (2002b) used word frequency to detect domain specificity of a term. They do not consider the label of neighbor terms to determine the label for a term.

2. Fukumoto and Suzuki (2011) have modeled domain labeling as a Markov Random Walk problem, but they run their algorithm on entire Wordnet graph. This is costly in terms of time and space required for the processing. In addition to that, Wordnet hypernymy-hyponymy graph may not be a true representative of domain specific conceptualization.

In contrast to the above mentioned approaches, our approach is based on the hypothesis that, 'Domain specificity of a term depends on the spatial property of the term'. So it is important to construct a domain specific conceptualization to identify domain of a term. The domain for a concept/term depends not only on the occurence of the term in the domain but also on the neighbors of the concept/term. Hence, we follow two step process in which first we construct a domain conceptualization from the corpus and then we align this conceptualization with Wordnet.

3 Algorithm

The proposed algorithm carves out a domain specific subgraph from the Wordnet. For that, we first construct concept graph from the corpus and then associate concepts with Wordnet senses. Figure 1 shows the overall system architecture. As shown in the figure 1 after preprocessing, the similarity graph is constructed from the corpus. Using a graph based algorithm similarity graph is converted into domain conceptualization and then it is linked with Wordnet synsets to assign domain labels to Wordnet synsets. The detailed description of each component is as follows.

3.1 Preprocessing

The text corpus is first POS tagged using Stanford POS tagger [1] and Morph Analyzer [2]. Then term frequency of each term is calculated using weirdness measure (Ahmad et al., 1999). Context vector for each term is constructed using Point Wise Mutual Information (Church and Hanks, 1990) measure. We used a sentence as a boundary to calculate context vector. Output of the preprocessing step is a list of domain specific terms and their context vector.

3.2 Constructing Document Graph

Using the term list and context vector generated from the preprocessing step, a graph G(V, E) is constructed in which each $v_i \in V$ is term and each edge $e(v_i, v_j)$ is semantic relatedness between terms v_i and v_j. Semantic relatedness between two terms v_i and v_j is calculated by taking cosine of terms vectors of v_i and v_j, as shown in fig 2.

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

Figure 2: Cosine Similarity

3.3 Constructing Domain Specific Conceptualization

Algorithm 1 Graph Cut Based Energy Minimization

```
  Input: set of labels $L$, undirected graph $G(V, E)$
  where, $V$ is set of random variables, $E$ is
  penalty cost, $f(v_l)$ is cost of assigning label
  $l \in L$ to $v \in V$, A set of initial labeling $\{(v, l)$,
  for all $v \in V$ and $l \in L\}$ and Energy Function
  $\theta$
  for $v_i$ and $v_j \in V$ do
    Source ← $v_i$
    Target ← $v_j$
    Perform Graph Cut
    Re-assign labels
    Calculate $\theta$
    Repeat until $\theta$ is minimized
  end for
```

[1] http://nlp.stanford.edu/software/tagger.shtml

[2] http://www.sussex.ac.uk/Users/johnca/morph.html

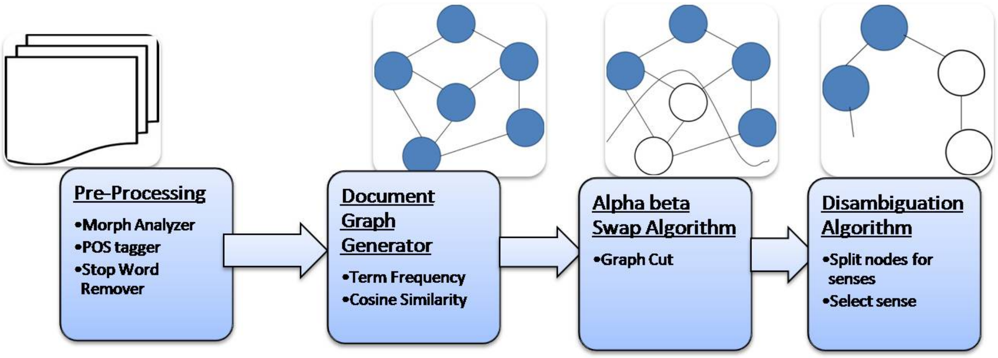

Figure 1: System Architecture

This module takes document graphs as an input and constructs a cohesive domain specific conceptual structure. In order to do this, we need to classify each node in the corpus graph into various domains. Assignment of a domain label to a node depends on two parameters,

- Term Cost: This measures how strongly a term belongs to the domain. It is measured by frequency of occurrence of a term within domain. This is formulated as a cost function as shown in equation 1.

$$tcost = \sum_{i \in V} E_i(X_i) \tag{1}$$

  where, X_i is the label assigned to term i and E_i is the cost of assigning label X_i to node i.

  We use term frequency based measure to calculate cost of assigning label to a term. A term should be assigned to a domain in which it occurs more frequently. Hence, high *tf* indicates less cost to assign the term to domain. Thus,

$$E_i(X_i) = 1 - tf_i \tag{2}$$

  where, tf_i is the term frequency of the term i in domain X.

- Edge Cost: This measures the cost of assigning separate labels to the two adjacent nodes of an edge. This is formulated as a cost function as shown in equation 3.

$$ecost = \sum_{(i,j) \in E} E_{ij}(x_i, x_j) \tag{3}$$

  where $E_{ij}(x_i, x_j)$ = cost of assigning different label to neighboring nodes i and j. $E_{ij}(x_i, x_j)$ is equal to semantic similarity between nodes x_i and x_j. Higher the similarity between nodes x_i and x_j more is the penalty to assign different labels to x_i and x_j.

This can be formulated as an energy minimization over a Markov Random Field (Kleinberg and Tardos, 2002). Finding optimal solution is equal to minimizing equation 4.

$$minimze\theta = \sum_{i \in V} E_i(X_i) + \sum_{(i,j) \in E} E_{ij}(x_i, x_j) \tag{4}$$

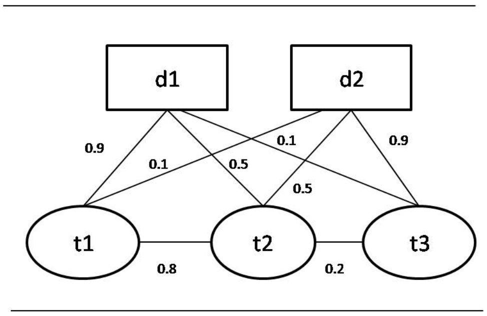

Figure 3: Domain Labeling

Figure 3 shows an example configuration of the concept graph with three nodes t_1, t_2 and t_3 and two domains d_1 and d_2. Edges from the nodes t_i to d_j indicates value of cost function $c(p, d)$ of equation 2. and edges between nodes t_i and t_j indicates cost for assigning different labels to node t_i

and t_j. As can be seen in the figure to minimize θ of equation 4, node t_1 will be assign to domain d_2 and node t_3 will be assign to domain d_1. Choice is to be made for t_2, since it has equal cost to be in d_1 or d_2. If t_2 is assigned label d_1, then $ecost$ of equation 2 is 0.8, since label for node t_1 and t_2 will be different. In the same way $ecost$ will be 0.2 if t_2 is assigned d_2. So to minimize θ, Final labeling is t_1 and t_2 are assigned d_2 and t_3 is assigned d_1.

In other words, to minimize the cost of assignment θ we cut the edge (t_2, t_3). Thus the energy minimization problem can be solved by performing 'Min-Cut' on graph. For two labels the problem is solvable in polynomial time. However, for more than 2 labels, solving this optimization problem is NP hard (Kolmogorov and Zabih, 2002).

In the field of image processing, many problems, *e.g. image forground-background detection, image segmentation etc.* are formulated as energy minimization in Markov Random Filed. Some of the graph-cut based algorithms to perform the task are, α-expansion, $\alpha - \beta$ swap (Schmidt and Alahari, 2011) and α swap β shrink algorithm . For our experiment we use α swap β shrink algorithm proposed in Schmidt and Alahari (2011). We are briefly describing the basic idea of the algorithms here. Readers are directed to Kolmogorov and Zabih (2002) and Szeliski et al. (2008) for further details.

For more than two labels (domains), a suboptimal solution can be derived by iteratively performing graph cut for a pair of labels. This problem is usually solved using iterative descent technique. As shown in algorithm 1, the algorithm start with an initial assignment. In each iteration the algorithm selects a pair of labels and performs the graph cut. Based on the graph cut the labels will be reassigned to the nodes. The energy function θ is calculated at the end of each iteration and the value of θ is minimized after every iteration to guarantee the convergence.

3.4 Split-Merge algorithm to Link concept to Wordnet

This module takes domain specific concept graph generated from previous step as an input and assigns wordnet sense to each term. A term can have more than one sense in the Wordnet and two terms can refer to same Wordnet synset. So the basic approach for the disambiguation is 'Split for Polysemy and Merge for Synonymy'.

Algorithm 2 *Link with Wordnet*

```
G(V, E)
V := vertices arranged in breadth first order
E := set of edges
$|V|$ := m $|E|$ := n
for $v_i \in V$ do
  create node $v'_i$ for each sense of $v_i$
  distribute edges across senses
end for
$v'$ := new sense vertex set; k := $|v'|$
for $i := 0 \to k$ do
  if Edge set $v'_i$ == 0 then
    delete $v'_i$
  end if
  for $j := 0 \to k$ do
    if Edge set $v'_i$ == Edge set $v'_j$ then
      merge $v'_i$ and $v'_j$
    end if
  end for
end for
```

As shown in Algorithm 2, the algorithm iterates through the nodes of the concept graph in a breadth first manner. For each vertex in the graph, all possible senses are found from the Wordnet. If a vertex v has n senses then new nodes v_1, v_2, ..., v_n are created. Then the sense nodes are linked with each other using Wordnet semantic relation, e.g. if two senses s_i and s_j are hypernym-hyponym in wordnet then and edge is created between them.

Figure 4 shows an example of vertex split. The left side of the fig. 4 shows concept graph for term node *cancer*. The term *cancer* has five different senses. Hence the algorithm creates five nodes for the term, one for each Wordnet sense. Then the edges are distributed across vertices depending upon the participating sense. Node *sign* is assigned to sense 1977832, and nodes *leukemia, lymphoma and Ailment* are assigned to sense 14239981. Other sense nodes do not have neighbors in the domain. Hence, sense 14239981 becomes winner sense in Health domain and it is tagged in the domain. Right side of the fig. 4 shows resulting wordnet sense graph.

Once new vertices are created for all vertices in the graph, the vertices with no edge are deleted and vertices for which the sense ids are same are merged as synonymy. Thus, at the end of the process we get a Wordnet sense graph specific to the domain. We label each sense with the specific do-

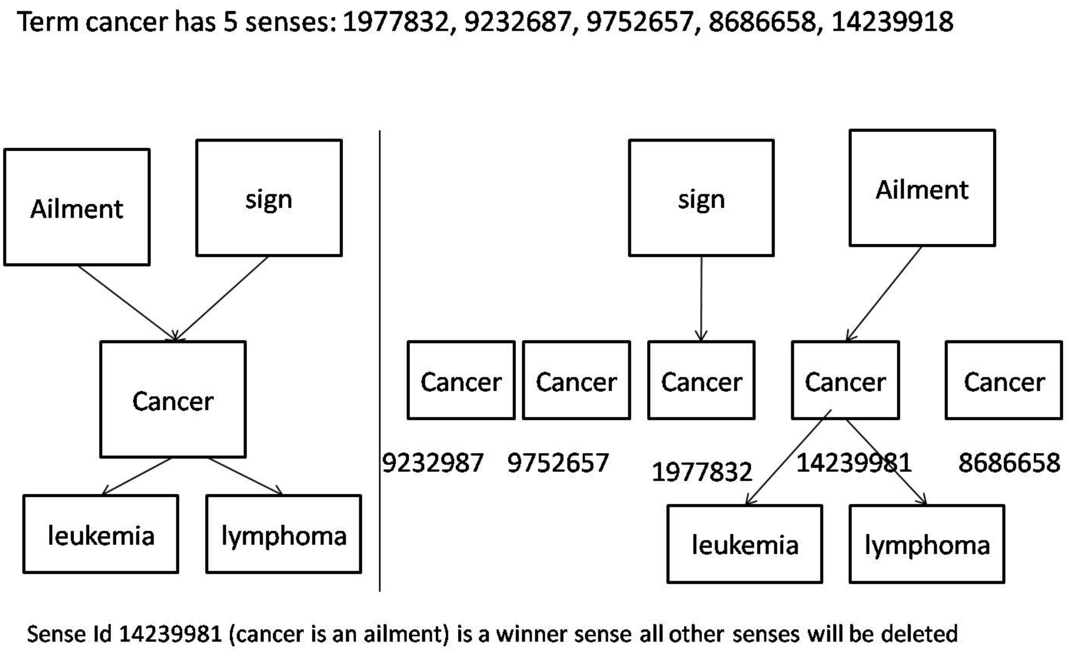
Term cancer has 5 senses: 1977832, 9232687, 9752657, 8686658, 14239918
Ailment
sign
Cancer
leukemia
lymphoma
sign
Ailment
Cancer
Cancer
Cancer
Cancer
Cancer
9232987
9752657
1977832
14239981
8686658
leukemia
lymphoma
Sense Id 14239981 (cancer is an ailment) is a winner sense all other senses will be deleted

Figure 4: Sense Splitting

	Health	Tourism
#terms	25056	56325
#terms after thresholding	4567	5968

Table 1: Corpus Statistics

Domain	Precision	Recall	F-Score
Health	0.69	0.82	0.74
Tourism	0.65	0.80	0.71

Table 2: Precision and Recall of domain labeling

main tag.

4 Experiments

We have conducted our experiments on publicly available Heath and Tourism Corpus [3] (Khapra et al., 2010). As shown in Table 1 the total number of unique terms after preprocessing and stop word removal are 25056 in health domain and 56325 in tourism domain. We applied further thresholding and remove low frequency terms (Frequency less than 10) to reduce the size of the graph.

For preprocessing we have used Stanford POS tagger and morpha morph analyzer. We have used Matlab UGM package [4] which is publicly available for researchers. UGM package provides implementation of α-expand, $\alpha - \beta$ swap and α-expansion-β-Shrink algorithms. The graph based disambiguation algorithm is written using JGraphT library [5].

The overall performance of the system is calculated against manually labeled domain tags. Table 2 shows overall precision, recall and f-score for both the domains.

As shown in Table 2 the recall value is found to be higher than the precision in both the domains. Reason for high value of recall is the initial labels and high number of edges. Initial labels are assigned based on the term frequency, then based on the labels of the neighboring nodes, node labels are changed. We observe that in case of two domains this leads to add more false positives. In order to reduce recall value and increase precision, we need to run experiments for more domains and with higher edge weights.

[3] http://www.cfilt.iitb.ac.in/wsd/annotated_corpus

[4] http://www.di.ens.fr/~mschmidt/Software/alphaBeta.html

[5] http://jgrapht.org/

5 Conclusion

We have proposed a novel graph based approach for automatic domain tagging of WordNet synsets. We pose domain labeling as an energy minization problem and show how the existing image labeling algorithms can be used for the task of WordNet domain tagging. Our approach is completely unsupervised and can be easily adopted across languages. For our experiments we used term frequency based assignment of initial labels, however other existing label can be used to enhance the labeling. In future we aim to construct domain labels for more domains and compare our system with existing labeling. We are also aiming to test our system for multiple languages.

References

Eneko Agirre and Aitor Soroa. 2009. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 33–41.

Khurshid Ahmad, Lee Gillam, Lena Tostevin, and Ai Group. 1999. Weirdness indexing for logical document extrapolation and retrieval (wilder). In *The Eighth Text REtrieval Conference*.

Luisa Bentivogli, Pamela Forner, Bernardo Magnini, and Emanuele Pianta. 2004. Revising the wordnet domains hierarchy: semantics, coverage and balancing. In *Proceedings of the Workshop on Multilingual Linguistic Ressources*, MLR '04, pages 101–108.

Paul Buitelaar and Bogdan Sacaleanu. 2001. Ranking and selecting synsets by domain relevance. In *proceedings NAACL wordnet workshop*.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29.

Philipp Cimiano and Johanna Vlker. 2005. Text2onto - a framework for ontology learning and data-driven change discovery. In *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB)*, volume 3513 of *Lecture Notes in Computer Science*, pages 227–238.

Fumiyo Fukumoto and Yoshimi Suzuki. 2011. Identification of domain-specific senses in a machine-readable dictionary. In *ACL (Short Papers)*, pages 552–557.

William A. Gale, Kenneth W. Church, and David Yarowsky. 1992. One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, HLT '91, pages 233–237.

Aldo Gangemi, Nicola Guarino, Claudio Masolo, and Alessandro Oltramari. 2003. Sweetening wordnet with dolce. *AI Mag.*, 24(3):13–24.

Aitor Gonzalez-Agirre, Mauro Castillo, and German Rigau. 2012. A proposal for improving wordnet domains. In *Proceedings of Language Resources and Evaluation Conference*, pages 3457–3462.

Mitesh M. Khapra, Anup Kulkarni, Saurabh Sohoney, and Pushpak Bhattacharyya. 2010. All words domain adapted wsd: finding a middle ground between supervision and unsupervision. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1532–1541.

Jon Kleinberg and Éva Tardos. 2002. Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. *J. ACM*, 49(5):616–639.

Vladimir Kolmogorov and Ramin Zabih. 2002. What energy functions can be minimized via graph cuts? In *Proceedings of the 7th European Conference on Computer Vision-Part III*, ECCV '02, pages 65–81.

Lung-Hao Lee, Yu-Ting Yu, and Chu-Ren Huang. 2009. Chinese wordnet domains: Bootstrapping chinese wordnet with semantic domain labels. In Olivia Kwong, editor, *PACLIC*, pages 288–296.

Bernardo Magnini and Gabriela Cavagli. 2000. Integrating subject field codes into wordnet. pages 1413–1418.

Bernardo Magnini, Giovanni Pezzulo, and Alfio Gliozzo. 2002a. The role of domain information in word sense disambiguation. *Natural Language Engineering*, 8:359–373.

Bernardo Magnini, Carlo Strapparava, Giovanni Pezzulo, and Alfio Gliozzo. 2002b. Comparing ontology-based and corpus-based domain annotations in wordnet. In *First International Global WordNet Conference, Mysore, India*.

Mark W. Schmidt and Karteek Alahari. 2011. Generalized fast approximate energy minimization via graph cuts: Alpha-expansion beta-shrink moves. *CoRR*, abs/1108.5710.

Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. 2008. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(6):1068–1080.

Ma Xiaojuan and Christiane Fellbaum. 2012. Rethinking wordnet's domains. In *Proceedings of 6th International Global WordNet Conference*, Matsue, Japan, jan.

Wang Yanna and Zhou Zili. 2009. Domain ontology generation based on wordnet and internet. In *Proceedings of the International Conference on Management and Service Science, 2009. MASS '09*, pages 1 –5.

Chaoyong Zhu, Shumin Shi, and Haijun Zhang. 2011. Gloss-based word domain assignment. In *7th International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE)*, pages 150–155.