Detecting Inter-domain Semantic Shift using Syntactic Similarity

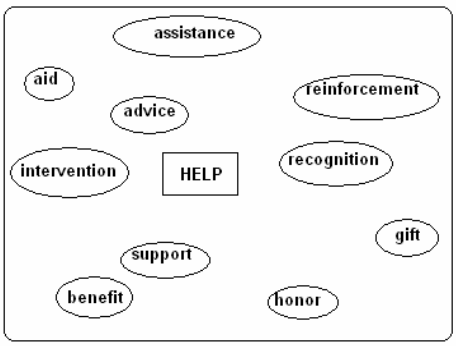
Masaki Itagaki, Anthony Aue, Takako Aikawa
Microsoft International Language Solutions, Microsoft Research - NLP, Microsoft Research - NLP
One Microsoft Way, Redmond, WA 98052
mitagaki@microsoft.com anthaue@microsoft.com takakoa@microsoft.com

Abstract

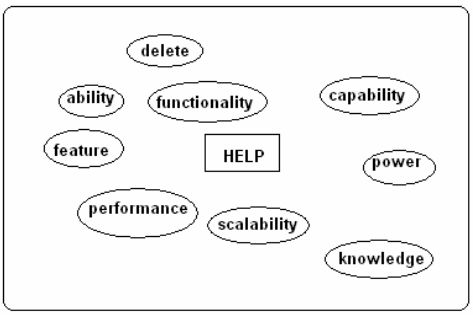
This poster is a preliminary report of our experiments for detecting semantically shifted terms between different domains for the purposes of new concept extraction. A given term in one domain may represent a different concept in another domain. In our approach, we quantify the degree of similarity of words between different domains by measuring the degree of overlap in their domain-specific semantic spaces. The domain-specific semantic spaces are defined by extracting families of syntactically similar words, i.e. words that occur in the same syntactic context. Our method does not rely on any external resources other than a syntactic parser. Yet it has the potential to extract semantically shifted terms between two different domains automatically while paying close attention to contextual information. The organization of the poster is as follows: Section 1 provides our motivation. Section 2 provides an overview of our NLP technology and explains how we extract syntactically similar words. Section 3 describes the design of our experiments and our method. Section 4 provides our observations and preliminary results. Section 5 presents some work to be done in the future and concluding remarks.

1. Introduction/Motivation

A word's semantic space varies depending on which domain it is being used in.[1] For instance, the semantic space associated with the word "help" in the technical domain is very different from its semantic space in the general domain, as shown in Figure 1 below.



General Domain



Technical Domain

Figure 1: The semantic space of the word "help" in the technical and general domains

Another such example is the word "stack." In our daily usage, it means a pile of letters, papers, etc. But in the technical domain, it refers to a first-in, first-out container for a set of objects. It is important for localizers to pay close attention to such semantically shifted terms when translating technical documents.

In this poster, we would like to explore a novel approach to extract semantically shifted words automatically, focusing differences between the technical and general domains. Although this poster focuses on these two domains, our ultimate goal is to implement a tool that can detect semantically shifted words between more subtly different domains, such as different versions of a product. This would allow for more efficient terminology management by localizers.

The high-level architecture of our approach is provided in Figure 2.

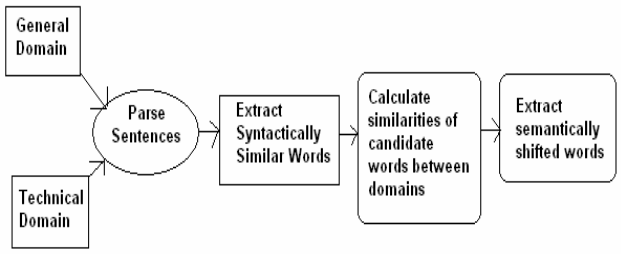


Figure 2: High-level Architecture

2. Syntactic Similarity

In our approach, the notion of syntactic similarity plays a critical role for defining the semantic space of a term. Our method for extracting syntactically similar words from a domain starts with parsing input data. We parse input sentences using a broad-coverage syntactic parser (Heidorn 2000). Based on the analysis of our parser, we generate a basic argument representation of the sentence, which we call its "logical form" (LF). Figures 3 and 4 show a parse tree and LF for the sentence, "Long data formats are not supported."

[1] Our usage of the notion of semantic space is similar in spirit to JurWordNet, which focuses on the semantic space of legal terms (see http://www.ittig.cnr.it/Ricerca/UnitaEng.php?Id=11&T=4). Our approach, however, differs from JurWordNet in that they rely on WordNet to extract semantically similar words, while we extract sets of syntactically similar words using our parser. See Section 2 for more details on syntactically similar words.

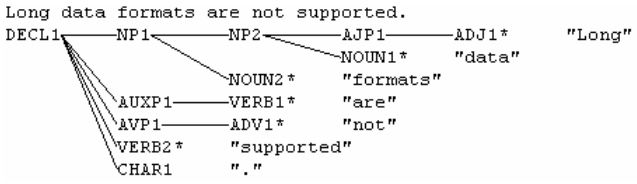


Figure 3: Parse Tree

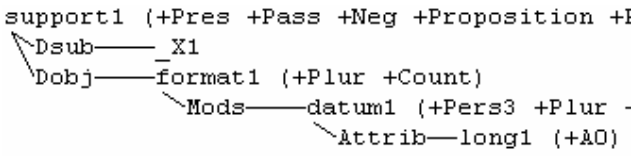


Figure 4: LF

As shown in Figure 4, we extract not only the basic argument structure of a sentence but also the types of the relationships between words. So for instance, the relationship between "long" and "data (datum)" in the above sentence is analyzed as an attribute (Attrib) and that between "long data" and the noun "format" is analyzed as a modifier (Mods).

We then extract a set of LF-triples from each sentence (i.e., <Word1, Relationship, Word2>) from the LF. Table 1 below shows the LF-triples extracted from the sentence above.

Word1	Relationship	Word2
support_verb	Object	**format**_noun
format_noun	Modifier	**datum**_noun
datum_noun	Attribute	**long**_adj
support_verb	Subject	**_X**_pronoun

Table 1: LF-triples for the sentence in Figures 3-4

Our method for extracting syntactically similar words from a domain is based on the approach described in (Lin, 1998). The basic idea of syntactic similarity is that if words occur in similar syntactic positions with respect to their heads or arguments, they are syntactically similar. For instance "bird" and "airplane" are syntactically similar in that they are both commonly the subject of the word "to fly." Note that syntactic similarity does not entail semantic similarity or synonymy. In fact, many words that are considered syntactically similar in our approach are antonyms (as well as synonyms). Table 2 provides samples of the syntactically similar words for "trade" and "early", using our method.

syntactically similar words for "***trade***"	syntactically similar words for "***early***"
investment	late
export	final
market	recent
economy	previous
industry	initial
sector	quick

Table 2: Sample examples extracted based on our method for "trade" and "early"

As shown in Table 2, "late" is considered as being similar to "early", despite the fact that it is an antonym of "early". Under our approach, we assume that a family of syntactically similar words can serve as a proxy for that of semantically similar words. Since we approximate semantic space in a domain by the set of words to which a given term is similar, we can use the degree of overlap between the two different sets of similar words from two different domains to determine how semantically similar the usage of the term is between the two domains.

3. Experiments

3.1. Data

We ran an experiment to determine the feasibility of our approach. The task was to identify terminologically interesting words between two domains: (i) the technical domain (a corpus of technical documentation) and (ii) the general domain (Canadian Parliament Parallel corpus [**Hansards**]). The sizes of the data used for this experiment are provided in Table 3. As described in Section 2, we parsed both corpora and extracted LF-triples of all the sentences.

	General Domain	Technical Domain
# of sentences	500,000	~4M (exact figure is 4,151,794)

Table 3: Training Data Sizes

3.2. Similarity Metrics

For our task it is necessary to quantify the degree of similarity for words between the two domains in order to extract terminologically interesting words. For our purposes, the less similarity in usage between the general domain and the technical domain, the more likely it is that a candidate is of terminological interest. In order to measure similarity of usage for a candidate between the two domains, we need to measure the amount of overlap between word families for the candidate between the technical domain and the general domain.

We applied Lin's algorithm to the sets of LF tuples (e.g. Table 1) from each domain in order to extract sets of similar words, with similarity scores, for each term in each domain. This can be seen as an approximate representation of the domain specific semantic space in which each candidate occurs. For each candidate, then, we have two sets of similar words, each with a score that indicates how similar the word is to the term candidate. For example, Table 4 below provides the partial results for the word "window" from the technical domain and the general domain.

Family of similar words for "window" in the technical domain	**Family of similar words for "window" in the general domain**
pane (0.135408)	door (0.0833651)
bar (0.115319)	glass (0.0723948)
frame (0.109951)	shop (0.0614379)
view (0.105979)	nose (0.0609436)
control (0.103643)	left (0.0592486)

form (0.102059)	link (0.0584969)
area (0.0990447)	sewer (0.0569417)
page (0.0984526)	delay (0.0567194)
icon (0.0978111)	wall (0.0555885)
screen (0.0972212)	avenue(0.0545522)

Table 4: Results for "window"

As one would expect, the similar words in the technical domain generally have to do with elements encountered in graphical user interfaces, whereas the most similar words in the general domain have to do with building and architecture.

In order to quantify the difference in semantic space for a term between two domains more precisely, we need a function that, given the word families and scores from the two different domains, returns a score indicating their degree of similarity. This score will allow us to rank terms by cross-domain similarity. In principal, the lower the cross-domain similarity for a term candidate, the more likely it is to be terminologically interesting.

We experimented with five different functions to determine the cross-domain similarity, some of which took the similarity scores provided by Lin's algorithm into account, and some of which did not. Manual inspection of the resulting ranking showed that, surprisingly, the simplest function was the best: simply count the number of words that overlap between the two sets.

For instance, for the word "window", there is no overlap between the syntactically similar words extracted from the technical domain and those extracted from the general domain (see Table 4). Thus, the semantic space of "window" in the technical domain and that in the general domain are different. Table 5 provides some of the words that have no overlap in their syntactically similar words and Table 6 provides those that have many overlaps.

shell	tag	sleep	envelop
crop	buffer	cell	garbage
partition	frame	sound	shutdown
icon	driver	void	explorer

Table 5: Samples of zero overlap items

reason	purpose	income	goal
challenge	concern	saving	idea
responsibility	expense	objective	tax
amount	dollar	factor	revenue

Table 6: Samples of many overlaps items

4. Results/Observations

We extracted about 367 terms (out of 1718 terms) that have zero overlap between the two domains. Here, we would like to list the family of similar words for "shell" and "tag" (see Table 5) so that the reader can see the contextual differences of these terms between the two domains in question and hence, can verify the semantic space shifts of these two terms between the two domains.

Family of similar words for "shell" in the technical domain	**Family of similar words for "shell" in the general domain**
debugger	partition
kernel	shot
installer	Clifford Olson
Windows Installer	ammunition
MMC	forestry
redirector	Brant
desktop	wound
subsystem	heck
Internet Explorer	bell
IDE	heroin
workstation	mining
Microsoft Outlook	adolescent
Windows Media Player	plum
runtime	sky
Terminal Services	salute
UI	tanker
debug	crab

Table 7: Families of similar words for "shell"

Family of similar words for "tag" in the technical domain	**Family of similar words for "tag" in the general domain**
element	SchoolNet
comment	Aida
block	NISA
declaration	SBLA
directive	Medical Research Council
html	Bill C-41
body	scholarship
header	TJF
symbol	irresponsibility
label	Parks Canada
markup	Crow
hyperlink	RRAP
mark	Small Business Loans Act

Table 8: Families of similar words for "tag"

As shown in Table 7 and Table 8, these two terms have totally different sets of similar words. We consider such terms as semantically shifted words between the two domains.

Unfortunately, the nature of our task makes it very difficult to evaluate the accuracy of our method objectively. However, manual examination of the results from our experiments is encouraging; in general, words with low intra-domain similarity scores are likely to be of terminological interest, and the converse is also true. We provide more examples of this sort at the poster session.

5. Concluding Remarks

In this poster paper, we explored an approach to identify semantically shifted words between two domains automatically without having any recourse to external synonym information (e.g., WordNet). We have shown that our approach, which is based on our NLP technology

and Lin's similarity metrics, can provide promising results for this task.

Although the paper discussed the results from the two extremely different domains only, in the future we plan to run experiments with two more closely-related domains in order to investigate whether we can detect subtle semantic shifts as well. We also realize that the lack of objective evaluation is problematic: manual evaluation of lists of words is expensive and error-prone. Therefore, we intend to carry out a more thorough evaluation against a database of known domain terms in order to quantify precision and recall. As mentioned at the outset of the paper, our ultimate goal is to implement a tool that can detect semantically shifted words for translators of technical documents so that terminology management can be done much more efficiently by localizers. We are in the process of utilizing this approach to capture semantically shifted terms from two different versions of the same product.

6. References

Dekang Lin (1998) Automatic retrieval and clustering of similar words. In Proceedings of COLINGACL '98, pages 768--774, Montreal, Canada.

Encarta (2005) http://encarta.msn.com.

Fellbaum, C. (ed.), 1998. WordNet: An Electronic Lexical Database, Cambridge, MA, MIT Press. Available at: http://www.cogsci.princeton.edu/~wn/.

Heidorn, G. (2000). Intelligent writing assistance. In R.Dale, H.Moisl and H.Somers (eds.), *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*. New York: Marcel Dekker, pp. 181-207.

Martin, L.E. (1990). Knowledge Extraction. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 252-262.