Chapter 1

Product Categorization

The two main tasks in this approach involve *identifying better features* for the product name and *evolving semantic measures of word similarity*. The feeling is that this will lead to features better than those obtained by treating the product title as a bag of words.

In the first phase, representative words called *focus words* and their *attributes* are extracted from the product title. This first phase is described in chapter 2. In the second phase, (semantic) similarity of these features with each product category is found. The product category that has the maximum semantic similarity with the *focus word* is output as the correct category.

The tools and parameters(distance/similarity matrices) for finding the similarity measure have been described in chapter 3. Please note that earlier, we had build the prototype using category names of just one word each (that was before we got your mail describing the product categories you use). We have not evolved a complete algorithm for handling categories consisting of multiple names. But our feeling is that based on our similarity measures, we could extend the algorithm to category names spanning multiple words.

So note that two main things need to be explored - using attributes in addition to the *focus words* and extending the similarity measure to finding similairy between pairs of multiple words or chunks.

Chapter 2

Query Processing

Let us consider a few sample product titles.

Softalk Shoulder Rests Model no 808ST BROIL KING TBS-3 Triple Buffet Server LAPTOP TRANSPORT CART W/POWER 35 1/2"WX21 1/2"DX38 1/8" 128MB Memory for Legend QDI KinetiZ 7T / n/a 3202-1950 Color Slim CD Jewel Cases By Memorex

Figure 2.1: Bag of words model

However, in each of these examples, we can see that some words in the title are more important for determining the product category than the others.

Softalk <**attribute**> **Shoulder** </**attribute**> <**focus**>**Rests**</**focus**> Model no 808ST BROIL KING TBS-3 <**attribute**> **Triple** </**attribute**> <**attribute**> **Buffet** </**attribute**> <**focus**> **Server** </**focus**> LAPTOP <**attribute**> **TRANSPORT** </**attribute**> <**focus**> **CART** </**focus**> W/POWER 35 1/2"WX21 1/2"DX38 1/8" 128MB <**focus**> **Memory** </**focus**> for Legend QDI KinetiZ 7T / n/a 3202-1950 Color Slim CD <**attribute**> **Jewel** </**attribute**> <**focus**> **Cases** </**focus**> By Memorex

Figure 2.2: Refined Model for product titles

From the figure 2.2, we see that some word(s) in the title can serve as the word of *focus* for the categorization task. For instance, in the first title above, the word of focus is "Rest(s)". However, we also find that the focus may sometimes not be sufficient for zooming down to a specific category and one may require more clue from the product title. Taking the same example, we find that "shoulder" is an attribute of "rest". Further, you might have multiple attributes for the word of focus.

2.1 Chunking with Focus word

To generalize the approach stated above, we may need to detect not just a *focus word* but *focus chunk*. This problem is related to the problem of *chunking* (refer *http://cnts.uia.ac.be/conll2000/chunking/*). Thus, "shoulder rests", "triple buffet server", "transport cart" and "memory" are *focus chunk*s.

Note that while detection of focus words is an extreme step away from the bag of words approach, detection of focus chunk is an intermediate approach. However, all words in the *focus chunk* are not equally important. For instance, in the chunk "transport cart", "cart" is more important than "transport" because you cannot categorize correctly with "transport" as the sole clue whereas "cart" as the sole key might suffice for categorization - only that the categorization will not be very fine-grained.

The attribute helps refine the categorization by disambiguating (refer chapter 3) the focus word. In a case above, "server" as a noun has 4 senses in WordNet [MBF+90]. However "triple" and especially "buffet" indicate that "server" is being used in its 4^{th} sense (utensil used in serving food or drink). This example further indicates the need for using attributes in disambiguating the noun.

2.2 Shallow parsing to extract focus word

An algorithm for detecting *focus word* in a title was used for the question answering system we developed and has been reported in [RCPB04]. Next, we describe a naive prototype we developed for focus word detection in a question. The only additional step required for focus word detection from product title is preprending the title with the cue "what is". For instance, the product title "128MB Memory for Legend QDI KinetiZ 7T / n/a" can be prepended with "what is" to give "What is 128MB Memory for Legend QDI KinetiZ 7T / n/a".

Shallow parsing, also called *partial parsing* or *chunking* [MS99, Page 375] involves finding noun phrases, modifiers, and attachments between phrases based purely on part-of-speech (POS) tags and without a deeper understanding of the content. For our purposes, shallow parsing is almost as robust as POS tagging, because we need only local attachments between verbs and noun phrases.

The parse tree that results from a shallow parse of the question *What is the capital of Japan?* is

```
(S
  (NP what)
   (S (VP is
          (NP (NP the capital)
              (PP of
                  (NP Japan))))))
```

Here *capital* is the "head" (the most important part) of the NP that is the sibling of the auxiliary verb *is*, and is the atype clue we need. Take another example:

What American general is buried in Salzburg?

whose shallow-parse tree is

```
(S
  (NP what American general)
   (VP is
       (VP buried
           (PP in
               (NP Salzburg)))))
```

The atype word is *general*, it has an attribute *American*, and these are in the NP before the auxiliary verb *is*.

This suggests a strategy:

1. If the head of the NP appearing *before* the auxiliary or main verb is not a wh-word, mark this word as an atype clue.
2. Otherwise, the head of the NP appearing *after* the auxiliary/main verb is an atype clue.

For question-answering, we find that this tiny rule set is surprisingly effective.

This algorithm was initially implemented using the *link parser* [ST93] which gives *dependency parse* of the question. However, we have now switched to the stanford unlexicalised parser [KM03].

A protype of the focus word extractor in java is present on our lab server at IIT, along with its GUI (based on servlets).

For the purpose of *product categorization*, we also tried modeling the problem of *focus word extraction* as a machine learning task. Of course, the features for learning were borrowed from the heuristics described above. The machine learning tools tried were decision tree and logistic regressor. Next we describe the corresponding machine learning approach.

2.2.1 Learning to detect the Focus Words

We have tried two set of features for training a classfier to detect the focus words in the question containing the title. We use WEKA's decision trees for the classification task.

For the first type of feature set, there is an instance for every word in the question/the title phrase. The features are :

1. Is the word under consideration, the head of a phrase
2. The phrase of which this word is a part. If the word is the head then this phrase is important
3. POS of the word
4. Phrase tag of the word after the current node
5. Phrase tag of the word before the current node

Results :

```
Accuracy of the decision tree classifier : 90.978%

=== Confusion Matrix ===

   a    b   <-- classified as
2080   74 |    a = n
 152  199 |    b = y
```

For the second type of feature set, there is an instance corresponding to every entry in the parse tree starting from depth 2. The features used for this classifier are : For every kth node in the parse tree of the question containing the title, at a depth of n, we have,

1. Depth of the entry in the parse tree *i.e.* n
2. The kind of phrase of the head of the entry
3. POS of the head of the entry
4. The kind of phrase of the previous *i.e.* (k-1)st node, at depth n
5. The kind of phrase of the next *i.e.* (k+1)st node, at depth n

Results :

```
Accuracy of the decision tree classifier : 81.3361 %

=== Confusion Matrix ===

   a    b   <-- classified as
1650  115 |    a = n
 332  298 |    b = y
```

We should however note that a single word (*focus word*) often will not suffice as an adequate clue for categorization and hence the output of the shallow-parse/focus word detector could be used to detect the attributes of the focus words as well - which could go into the next phase of the algorithm - finding similarity between the focus word/chunk and the product categories.

Chapter 3

Similarity Measures

The task is to find similarity between two words-senses. Each word-sense consists of an associated part of speech tag and a sense number. For words present in WordNet [MBF+90] , the sense number corresponds to WordNet sense number. Words not from WordNet will default to having only one sense. How this is done will be described later in this section.

There are two parameters for finding similarity between a pair of words - the type of gloss used and the similarity measure used.

Before we venture further, we first describe what is the meaning of a word-sense. A word can have multiple parts of speech and it is used in a sentence with a unique part of speech. This is one syntactic aspect of disambiguation at word level. Another level of disambiguation for a word, with its associated part of speech is the meaning it carries in the context it is used in. The word "tape" for instance, could be used as either a *noun* or a *verb*. As a noun, it could be used with different meanings. The noun "tape" has 5 meanings according to WordNet [MBF+90] - *viz.*

1. a long thin piece of cloth or paper as used for binding or fastening
2. a recording made on magnetic tape
3. the finishing line for a foot race
4. measuring instrument consisting of a narrow strip (cloth or metal) marked in inches or centimeters and used for measuring lengths
5. memory device consisting of a long thin plastic strip coated with iron oxide; used to record audio or video signals or to store computer information

In a title such as "Tatco Aisle Marking Hazard Tape, 2"x108', Yellow/Black", tape is a noun and is being used in the 4^{th} sense. On the other hand, in a title as "Onstream SC50-01 50GB Internal SCSI Digital Tape Drive", "tape" is being used as a noun, in its 5^{th} sense.

3.1 Glosses

Descriptive glosses

A word, with its associated part of speech and an associated sense number, has a description.

Description for fifth noun sense of "tape" memory device consisting of a long thin plastic strip coated with iron oxide; used to record audio or video signals or to store computer information

Figure 3.1: An example *descriptive-gloss* for "tape" from WordNet

Description for first noun sense of "case" a portable container for carrying several objects

Figure 3.2: An example *descriptive-gloss* for "case" from WordNet

We call these descriptions as *descriptive glosses*. For word-senses picked up from WordNet, the WordNet glosses are the descriptive glosses. WordNet glosses also contain example usages of the word. We have dropped out examples from the descriptive glosses. They could also be included.

For other word-senses, the descriptions could come from glossaries (like glossaries of software terms), encyclopedias (for names of people, places, events, pacts *etc*), world fact books, abbreviation lists *etc*. Examples glosses picked up from above sources are listed below.

Hypernymy glosses

The gloss for a particular sense of a word could also describe what hierarchical categories it belongs to.

For instance, the hierarchical categorization of the fifth noun sense of the word "tape" is

```
=> memory device, storage device
    => device
        => instrumentality, instrumentation
            => artifact, artefact
                => object, physical object
                    => entity
                => whole, whole thing, unit
                    => object, physical object
                        => entity
```

The hierarchical categorization of the 7^{th} noun sense of the word "case" is

```
=> container
    => instrumentality, instrumentation
        => artifact, artefact
            => object, physical object
                => entity
            => whole, whole thing, unit
                => object, physical object
                    => entity
```

The hierarchical categorization of the 1^{st} noun sense of the word "vesuvius" is

Description for first noun sense of "Volcano" a fissure in the earth's crust (or in the surface of some other planet) through which molten lava and gases erupt

Figure 3.3: An example *descriptive-gloss* for "volcano" from WordNet

descriptive-gloss for "Tuareg" A white-skinned nomadic people of the Sahara who speak an Hamitic Berber

descriptive-gloss for "piccolo" an instrument of the woodwind family. Most of these instruments were once of made of wood, and because they are played by blowing with air or wind, they are called woodwind.

descriptive-gloss for "DARPA" The Defense Advanced Research Projects Agency is the central research and development organization for the Department of Defense in the U.S.

Figure 3.4: Examples of descriptive glosses for non-WordNet words picked from glossaries

```
=> volcano
      => mountain, mount
          => natural elevation, elevation
              => geological formation, formation
                  => natural object
                      => object, physical object
                          => entity
```

Based on this hierarchical categorization of the first noun sense of "Vesuvius", we describe its *hypernymy-gloss* as the collection of all nodes in its hypernymy-path to the root - *viz.* "entity".

Hypernymy gloss for first noun sense of "Vesuvius" {volcano}, {mountain, mount}, {natural elevation, elevation}, {geological formation, formation}, {natural object}, {object, physical object}, {entity}

Figure 3.5: Hypernymy-based gloss first noun sense of Vesuvius

Whereas *descriptive-glosses* can be derived even for word-senses not present in WordNet, *hypernymy-glosses* require classification of word-senses into nodes into an ontological structure - like the hypernymy hierarchy of WordNet. This is not that easy to procure for words not present in WordNet. One possible solution, and the one that we actually resort to, is to find the named entity tag for a token (if one exists) and then map the tag to a node in WordNet. For example, the token "President Musharraf" is not present in WordNet. But this token can be tagged as a *PERSON* and *PERSON* could be mapped to a node in WordNet - *viz.* the first noun sense of "person" (person#n#1). Similarly, the token "26^{th} December 2003" has a DATE named-entity tag. DATE could be translated to the 7^{th} sense of the word "date" (date#n#7) in WordNet.

Thus, *hypernymy-glosses* of many words not present in WordNet (which currently default to having only one sense) could be evolved from their named-entity tags.

Another parameter for measuring similarity between two word-senses is the similarity metric. Given two word senses, once we pick their glosses, we need to find similarity between the word senses, based on the glosses choses. We next describe different similarity metrics.

3.2 Similarity metrics

The similarity between two word-senses, for a given gloss is found by tokenising each gloss, constructing vectors from the two tokenised glosses and finding similarity between the vectors.

Cosine similarity

One standard metric of similarity, as used in information retrieval, is the cosine-similarity. We find the cosine similarity between the *term frequency-inverse gloss frequency (tfigf)* vectors of the two glosses. The *inverse gloss frequency (igf)* of a token is the inverse of the number of glosses which contain that token and it captures the "commonness" of a particular token.

There have been fancier definitions of similarity in literature [Lin98] which involve information theoretic measures of similarity between word-senses, based on the hypernymy path and DAG structure of WordNet. These methods are heavily dependent on frequencies of synsets in a sense-tagged corpus. The idea is that two word-senses are highly related if their subsuming synsets are highly information bearing - or in other words, have high information content. Information content is computed from a sense tagged corpus - word-senses with *high* frequencies of occurrence have *low* information content. This brings in the the problem of data sparsity - because sense-tagged corpora are very scarce and of short size. Their coverage of synsets is poor as well. Hence there is the danger of making the similarity measure biased toward the sense-tagged corpus.

Also, these methods are very slow and CPU intensive, since finding similarity between two word-senses at run time involves traversing the WordNet graph, in the direction of hypernymy links, up to the least common ancestor.

On the other hand, a cosine similarity on tfigf vectors built from *hypernymy-glosses*, gives a low similarity value between word-senses whose hypernymy-glosses overlap in very frequently occurring synsets relative to the synsets which are not common to their glosses. This is because *igf* implicitly captures the information content of a synset - the higher the igf - higher is the information content of a synset. The purpose served by a sense-tagged corpus is cumulatively served by the collection of hypernymy glosses of all the WordNet synsets. This method is also more reliable since the *igf* values come from WordNet which is very exhaustive, unlike sense tagged corpora (like SemCor) which will have bias and data-sparsity in terms of which words occur in the corpus and which sense is picked for a word. (The reader might want to note some work which has been done to illustrate that words can inherently have multiple senses in a given context).

The cosine similarity on tfigf vectors built from *descriptive glosses* is very much like the similarity found between document and query vectors, since the tokens in descriptive glosses are regular words. Cosine similarity is intuitively the most useful similarity measure on descriptive glosses since cosine similarity of *tfigf* vectors takes care of stop words and very non-informative words like "the" *etc.*

Jaccard similarity

Another metric of similarity is the *jaccard similarity*. Jaccard similarity between two sets of tokens (glosses) is computed as $\frac{|A \cup B|}{|A \cap B|}$. Here A and B are the two glosses.

Jaccard similarity is appealing only if the glosses used are *hypernymy-glosses*.

Asymmetric measures of similarity

The above two were symmetric measures of similarity. A third asymmetric similarity measure is one that takes a value of 0 if the intersection of the glosses of two word-senses is not equal to the gloss of one of the word-senses. Else, the similarity is equal to one of cosine or jaccard similarity measures. This means that there are actually two asymmetric similarity measures - one due to jaccard and the other due to cosine.

Bibliography

[KM03] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the ACL Conference*, 2003.

[Lin98] D Lin. An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pages 296–304, San Francisco, CA, 1998. Morgan Kaufmann.

[MBF+90] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: an on-line lexical database. *International Journal of Lexicography 3 (4)*, pages 235 – 244, 1990.

[MS99] Christopher D Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.

[RCPB04] Ganesh Ramakrishnan, Soumen Chakrabarthi, Deepa Paranjpe, and Pushpak Bhattacharyya. Is question answering an acquired skill ? In *Proceedings of the 13th World Wide Web Conference (WWW13)*, 2004.

[ST93] D. D. Sleator and D. Temperley. Parsing English with a link grammar. In *Third International Workshop on Parsing Technologies*, 1993.