Mining Grammatical Categories

June 2015

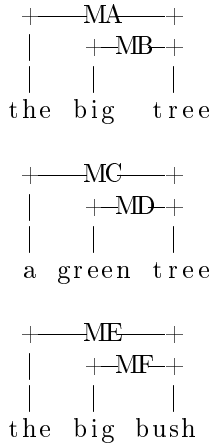
20 June 2015

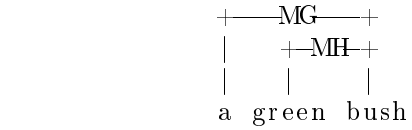
Now that we have a database filled with disjunct statistics, how do we datamine that for grammatical categories, which is, after all, the main point of this exercise? Let me explain in several steps; at first illustrative, and then, more precisely. So first, consider a corpus containing these sentences:

the big tree
a green tree
the big bush
a green bush

I want to conclude that "tree" is a lot like "bush", and the two should be considered as being "similar enough to be merged into a common grammatical category". That is, the words "tree" and "bush" always occur in similar contexts, or even the same contexts. The word "context" here means "the dependency parse context", and not "the n-gram context". More precisely, it means "the accumulated statistics for the disjuncts obtained from MST dependency parses".

Suppose the following parses were observed:





Recall that the above parses were obtained by performing a Maximum-Spanning-Tree (MST) parse based on word-pair mutual information (MI). The MST is obtained by considering the graph clique joining all words in the sentence, and then keeping only those edges that have the greatest MI between pairs of words. This is the "Yuret parse". The Yuret parse does not have labelled edges, and so we assign arbitrary (but unique!) link labels to the edges that were kept. Every unique word pair gets a unique link type. Then, using the standard Link Grammar theory, each link is broken into a + and a - connector, and the ordered set of connectors on a word are called a disjunct.

The disjuncts extracted from the above parses would then be:

tree: (MA– & MB–) or (MC– & MD–)
bush: (ME– & MF–) or (MG– & MH–)

No two disjuncts are alike, so naively, these seem completely uncomparable. Of course, this is wrong; we need to compare the "decoded disjunct". The "decoded disjunct" is NOT a part of the standard Link Grammar theory, so let me explain it here: it is simply the disjunct where the connector is replaced by the word or word-class that it connects to. For example, `MA-` connects to the word "the", so the "decoded connector" for `MA-` is `$the$-`. So, the decoded disjuncts are then:

tree: (the– & big–) or (a– & $green$–)
bush: (the– & big–) or (a– & $green$–)

Now we can see that the decoded disjuncts are identical, for this example. Based on this, we conclude that perhaps "tree" and "bush" indeed belong to the same grammatical category. The remainder of the clustering algorithm is now "obvious": rewrite the dictionary so that it has a single entry for both words:

tree bush: (MA– & MB–) or (MC– & MD–)

This leaves the ME+, MF+, *etc.* connector dangling: thus, we need to search for all occurances of ME+ and replace it by MA+, and likewise all occurances of MF+ need to be replaced by MB+, and so on.

Similarity metrics

The above conveys the general idea, but is over-simplifies a few aspects. First of all, it is very unlikely that two words will appear in sentence contexts that are exactly identical. Secondly, some constructions may be very common, and others, very rare; that is, some disjuncts may be very common, and some very rare. So, for example: suppose we read a text which used the phrase "*the big idea*" a lot, but we also read an obscure linguistics text that said that "*a green*

idea sleeps furiously". It would probably be a mistake to lump "idea" in with "tree, bush", given that "green idea" is a very rare construction. Thus, we need a better way of comparing collections of disjuncts.

One obvious way is to treat a collection of decoded disjuncts as a vector in a high-dimensional vector space. The similarity between two vectors could be given by the cosine between two vectors. Alternately, perhaps the vectors could be treated as points, and similarity be given by the distance between points. There are other possibilities; the best choice is not obvious; several need to be explored.

Thus, for example, let $\{e_1, e_2, e_3, \cdots\}$ be the basis of a high-dimensional vector space. For the previous example, we let e_1 correspond to the decoded disjunct `($the$- & $big$-)` while e_2 corresponds to `($a$- & $green$-)`. The word "tree" is then some vector ... what vector should it be? There are several choices. Suppose that `($the$- & $big$-)` was observed with a frequency p_1 and that `($a$- & $green$-)` was observed with frequncy p_2. The corresponding vector is then obviously $p_1 e_1 + p_2 e_2$ and we can construct another vector that corresponds to the the word "bush", say, for example: $q_1 e_1 + q_2 e_2$.

The dot-product between "tree" and "bush" is then given by $p_1 q_1 + p_2 q_2$, so that the larger the product, the closer the two words are. The cosine angle is $(p_1 q_1 + p_2 q_2)/\,|p|\,|q|$ where $|p| = \sqrt{p_1^2 + p_2^2}$ and so on. The closer that the cosine is to 1.0, the closer the two words are. There are other possibilities: we have the Cartesian distance

$$dist(tree, bush) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$$

and we can contemplate *lp*-metrics as well.

None of the above metrics take into account the mutual information (MI) of the disjunct. This is almost surely a mistake. Due to the vagaries of MST parsing, there will be many disjuncts with a low MI value. This is not uncommon in sentences with prepositions, where MST gives some poor choices for the links to the prepositions, and thus results in disjuncts with low MI values. Recall, the higher the MI, the stronger the structure is. Thus, perhaps a better vector for "tree" might be

$$tree = e_1 m_1 p_1 + e_2 m_2 p_2$$

The above seems to be the most entropic-like in its expression. However, the probabilites might weight the terms too strongly, and so a weaker weighting would be the below. It is not yet clear to me which of these expressions are the most "elegant", or which work the best...

$$tree = e_1 (m_1 - \log_2 p_1) + e_2 (m_2 - \log_2 p_2)$$

Here m_1 and m_2 are the mutual information of the disjuncts `(MA- & MB-)` and `(MC- & MD-)`, respectively. The last two seem to be closer to the intended spirit of the maximum entropy principle. There are even more possibilities, though.

Frequency and Mutual Information

The above section makes explicit use of the frequency and the mutual information of a disjunct. It is useful to define these. Given a disjunct `(MA- & MB-)` let N`(MA- & MB-)` be the number of times that this disjunct has been observed. It will usually be an integer (except when obtained in certain unusual situations not discussed here). Let N`(*- & *-)` be the numer of times that any two-connector disjunct has been observed, as long as both connectors point in the - direction. That is,

$$N(*- \& *-) = \sum_{c_1 \in -, c_2 \in -} N(c_1 \& c_2)$$

the summation taking place over all connectors in the - direction. The frequency of observing `(MA- & MB-)` is then

$$p(MA- \& MB-) = \frac{N(MA- \& MB-)}{N(*- \& *-)}$$

The mutual information associated with the disjunct is then

$$m(MA- \& MB-) = -\log_2 \frac{p(MA- \& MB-)}{p(MA- \& *-)p(*- \& MB-)}$$

The reason for this possibly unexpected form was developed earlier in this diary.

Semantics

There is another interesting issue that arises in the above discussion: the problem of syntax-semantics correspondance. Consider, for example, the sentence *"the dog treed the squirrel"*. Here the word "tree" is used as a verb, meaning "the dog chased the squirrel up into the tree". Such sentences will cause the the word "tree" to accumulate disjuncts that the word "bush" will not have. Likewise, "*I'm bushed*" is a verb usage that has no analogous "tree" version. Thus, not only do the words "bush" and "tree" have different sets of disjuncts, but the differences are hiding semantic differences ...

There are several strategies that can be used to deal with this. More on this later.

Finding word pairs

We need a good way of finding word-pairs that are likely to be related. I think that perhaps the pattern matcher may be ideal for this. Details are TBD... but the basic idea is that the hypergraph for "tree: (MA- & MB-)" is connected to "big" because MB- is connected to "big", and "big" is connected to other lg-connectors, which in turn are connected to other disjuncts, which are then connected to other words. Thus, we search the local neighborhood of "tree", which causes us to dsicover the word "big", and then we search the neighborhood of "big" to find candidates such as "bush" which might be comparable to "tree". This search graph is not small, but it is not large: There may be thousands of words that are two hops away from "tree", but not millions.

Putting it all together

These are the things that need to be done:

1. compute the MI for the disjuncts
2. pick a common noun, compute the similarity scores for that word and every word that is linked to it. created ranked graph of similarity.
3. repeat step 2 for several different similarity formulas
4. repeat steps 2,3 for several verbs, several adjectives, several adverbs, several determiners, several prepositions.
5. Write code for creating grouping words into grammatical clusters.
6. Pick the most promising metric, and start clustering in bulk.

Step 5 requires writing a lot of code; it can all be written before the final metric has been determined.

The end.

That's all for now. More later.