Emily Mulhall
COMP 550
Assignment 3
November 13, 2017

Question 1

1. (a) $(\lambda x.xx)(\lambda yx)z$ to $(\lambda y.yx)(\lambda y.yx)z$ to (λ y.yx)xz to xxz

   (b) $(\lambda uvw.wvu)aa(\lambda pq.q)to(\lambda pq.q)aatoa$

   (c) ((λ v.vv) (λ u.u))((λ v.v)$(\lambda v.w))to(\lambda u.u)(\lambda v.w)\lambda v.w$

2. Please see the scanned pagess for the derivation of *No student hates COMP-550*

3. Please see the scanned pages for the derivation of *No student wants an exam.* The two possible readings are: there does not exist any student which wants any exam and there does not exist any student which wants a particular exam under discussion.

Question 2

In this question three models were compared: a baseline model, a Lesk model, and a model combining information about the distribution of senses and the Lesk algorithm. Unfortunately, not a single model got above 62% on the test instances. In fact, the baseline model performed the best. This is highly surprising, as the Lesk algorithm is designed for this task.

The Lesk model, which compares the overlap of the context words of each synset resulting from the target word with the word's definition, achieved an accuracy of about 30% on the task. One of the difficulties this model faced was the fact that many of the instances had a big number of synsets to choose from. A decent amount of instances had over 20 to choose as a sense, and more than one had 40 to choose from! Because these synsets are related to the ambiguous word, it's likely that much of their context will overlap with the target word's definition. With that many possible synsets, it is highly likely that the amount of overlapping will be quite close in many of them. This could easily result in the wrong sense even if the correct sense context overlaps with the target word definition significantly. It is also possible that a tie will result between more than one sense, and even if the correct sense is one of the senses with the highest amount of overlap, it still might not get chosen. Additionally, it is entirely possible that the context words of the correct sense simply do not overlap much with the definition of the ambiguous words. The words in the definition of a target word might not always be exactly the context words of the correct synset. The context words might be closely related to the words in the definition, but that does not mean that there will be a high amount of overlap of exact words. Thus, there are many reasons that the Lesk model will fail. However, there are also many cases that the correct sense will have context words that highly overlap with the definition of the ambiguous word which is why 30% is achieved.

Unfortunately, many of the previously stated reasons also make it very difficult for the modified Lesk algorithm. The modified Lesk model combines the Lesk algorithm with distributional information regarding senses. In this model, if there is only one possible synset, then that is returned as the sense. However, it also takes into account an additional parameter, data. Data is a list that records every sense that has been assigned to an instance so far. If this list is empty, then the algorithm reduces to the Lesk algorithm; it simply compares the overlap of the context words of each synset with the definition of the ambiguous word. If data is not empty, however, the algorithm uses frequency information of senses. A frequency distribution is first created from the synsets that are possible from the ambiguous word. If any of these synsets have already been used as a sense for a previous instance, then the amount of times it has been used as a sense is added to its frequency in the distribution. The maximum likelihood estimate (MLE) of the frequency is taken, and the synset with the highest probability is returned as the sense. Thus, this model makes senses which have already appeared for previous instances more probable. However, many senses do not appear at all or more than once, and thus this additional information does not always help in choosing a sense. Yet, the modified model was able to achieve a slight improvement in accuracy over the unmodified Lesk Model, managing about a 31% accuracy. This small increase could be explained by the possibility that some senses are simply more common than others. Because the algorithm chooses senses that have already been seen, it takes advantage of the fact that rarer senses are less likely to appear in the dataset. An additional difficulty faced by this model is the fact that the data that makes up the test set does not all come from the same source. Thus, it is possible that while some senses are more common in one source, they are less common in another source. If this is the case, then there are instances when the model will choose the wrong sense because it had already seen it in a different source while it had not seen the correct one.

The baseline actually performed the best, achieving about 62% accuracy on the test data. The baseline simply chose the first synset in the set of synsets made possible by the ambiguous word as the sense. Thus, it is quite surprising that the baseline achieved an accuracy that more than doubled both the accuracy of the modified and (unmodified) Lesk models. This high accuracy is most likely due to the fact that the most common sense of a word is listed first in its set of synsets. Because the more common senses appear extremely frequently, the baseline was able to choose the correct sense more than half of the time.

Overall, I was quite disappointed with the performance of the modified Lesk algorithm and the Lesk algorithm. Perhaps if one sense has already been chosen for a previous instance, rather than immediately increasing its likelihood its context should first be compared to the definition of the ambiguous words and only if it significantly overlaps with the definition should its likelihood be increased. I additionally believe that more data in the test set would lead to a greater improvement in accuracy for the modified Lesk algorithm. However, this would most be unable to surpass the accuracy of the baseline, as the modified Lesk algorithm combine the ideas of the Lesk model and the baseline model. It does choose the first instance using the Lesk algorithm, but then it begins to look for more commonly used senses. Thus, it is possible that with additional information, while the accuracy of the modified Lesk algorithm will improve, it might not perform much better than the baseline.

Question 3

In this paper the authors compare multiplicative and additive vector-based models of words on a sentence similarity task. The task involved ranking the similarity of the meaning of one sentence with the meaning of a reference sentence. To keep the task simple the authors used sentences containing only a subject and an intransitive verb. They also asked humans to perform this task and compared the results of the models to the results of the human participants.

Overall, the general comparison of additive and multiplicative models was a key focus of the paper. For the additive models they had a simple additive one ($p_i = u_i + v_i$), a weighted additive one ($p_i = \alpha u_i + \beta v_i$), and a model proposed by Kintsch in 2001 ($\mathbf{p}=\mathbf{u} + \mathbf{v} + \ \Sigma\ \mathbf{n}$). Kintsch's model involves the main idea of distributional semantics, which is that a word's meaning is related to the words that are often in its context. By including the neighbors surrounding the target word, Kintsch hoped to better capture its meaning. The other two main models of focus were the simple multiplicative one ($p_i = u_i \cdot v_i$) and the combination one ($p_i = \alpha u_i + \beta v_i + \gamma u_i v_i$). The multiplicative model is often believed to be a less accurate representation as compared to the additive ones because if one of the vectors contains a zero, then it will eliminate information when multiplied with the other vectors representing the rest of the sentence. The combination vector is included in the comparison in order to capture the effect of multiplication and maintain all possible information. Yet, the difference between the multiplication model and the combination one was not significant which results in the question: if the combined model uses the multiplicative model in combination with the additive one in order to prevent the loss of information, then why are the results of this model and the multiplicative model not significantly different? In fact, the multiplication and combination models performed far better than any of the additive ones. The authors explain this result with the conjecture that additive models are not sensitive to the fine-grained meaning distinctions. But why aren't they sensitive to these distinctions? Or at least, why are they less sensitive than multiplicative models? It was also surprising that the Kintsch model, the only one that attempts to include context, did not perform very well. Why is it that a model that includes the main idea of distributional semantics performs worse than models that to do not?

This paper covers many topics covered in class including distributional and compositional semantics, vector representations of word meanings, and word senses. I would be interested to see these models compared on another semantic task, such as an entailment and contradiction classification task or on more complex sentences. I believe the syntactic information would be more important in these tasks (based on the main idea of compositional semantics), and it would be interesting to see if the weighted additive model performs better than the others on them.