# Summary Report (08/28/2014)

## Overview

In this week, I have done the following things:

* Different Semantic metrics
* MMR ranking
* Adjective Pair

## More Semantic Metrics

Due to a poor documentation of the SEMILAR toolkit, not all of the metrics are well defined.

* greedyComparerWNLin

Greedy matching: assume the similarity of a word to a sentence is the maximum similarity score from the word to any word in the sentence.

If use wordnet LIN method for Word 2 Word similarity:

* optimumComparerLSATasa

Optimal Matching: instead of using greedy matching, it is based on the optimal assignment problem (maximum weighted bipartite graph matching). Different words will be matched to different words.

It uses LSA based word 2 word similarity (using TASA corpus LSA model).

* optimumComparerWNLin

See paper Corley, C. and Mihalcea, R. (2005). Measuring the semantic similarity of texts. It uses the Word Net Lin word 2 word similarity.

* dependencyComparerWnLeskTanim
* bleuComparer

Used the BLEU score from machine translation.





B is a brevity penalty,

* cmComparer

METEOR method (introduced for machine translation evaluation): http://www.cs.cmu.edu/~alavie/METEOR/

* lsaComparer
* lexicalOverlapComparer

The lexical overlap without using the base form of words

### Results:

The BleuComparer, cmComparer and LSAComparer don’t give me correct similarity score (always predicting NAN), so their results are ignored (It is hard to figure out what’s going under the hood because the source code is not available).

In addition, because all the similarity scores given by SEMILAR are normalized to 0~1, it will not favor to longer phrases. Thus, for each of the similarity in SEMILAR, a new similarity is proposed as follows:

Note: it is asymmetry because it is normalized by the length of the first phrase.



* Generally speaking, the length-normalized version of similarities are better than the original ones. (Highlighted Green cells are better ones.)
* The greedyComparerWNLin is the best one for POI among the Semantic similarities; optimumComparerLSATasa is the best for MP and LP
* However, none of them are better than NP-Soft
  + One possible reason is that the semantic similarity is used only to compute the score of the phrases, but not used to remove the duplication. (The duplication removing is still based on the stemming introduced in the 08-07-2014 report)
  + Another reason is the extracted noun phrases are not at the right granularity. For example, the phrase “what we will learn in this class” will be split to NP “what we”, “this class”. In this case, it is hard to capture the semantic meaning. In addition, the proposed semantic similarities given by SEMILAR are for words or sentences.

## MMR ranking (duplication removing)

Recall that, in the current shallow-based extractive summarization (refer to report 08-07-2014 for detail), I maintain two sets: the selected sentences and the word set. If a word is selected in the sentence, it will be removed from the word set in order to remove duplication.

As shown by Wenting, another common way to handle duplication is MMR [2]: Maximal marginal relevance. The basic idea is to select sentences are not only relevance/ representative, they should also be different from each other (contain new information).

The definition is given below:

It is used in Information retrieval (Q is the query and D are the documents). Adapt to automatic summarization, the second term becomes to the similarity between the extracted sentences.

Fortunately, the MMR method is already implemented in Mead.

### Result:

In the implementation, ranges from 0 to 1 with 0.2 step. The I use is cosine.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | POI | | | MP | | | LP | | |
|  | R1 | R2 | R-SU4 | R1 | R2 | R-SU4 | R1 | R2 | R-SU4 |
| default mead | 25.07% | 8.33% | 5.43% | 25.03% | 8.51% | 6.23% | 14.47% | 1.72% | 2.28% |
| unigram-based | 26.71% | 9.02% | 5.57% | 31.39% | 11.60% | 7.97% | 15.46% | 1.35% | 2.18% |
| topicS-based | 28.53% | 9.70% | 7.18% | 31.80% | 9.78% | 8.70% | 16.04% | 1.56% | 2.27% |
|  |  |  |  |  |  |  |  |  |  |
| mmr\_lambda\_0 | 22.37% | 7.08% | 4.03% | 21.33% | 5.52% | 4.52% | 14.48% | 1.28% | 1.78% |
| mmr\_lambda\_0.2 | 22.69% | 7.08% | 4.09% | 22.65% | 5.88% | 4.75% | 14.49% | 1.26% | 1.79% |
| mmr\_lambda\_0.4 | 22.87% | 7.08% | 4.06% | 23.39% | 5.94% | 5.16% | 14.02% | 1.07% | 1.75% |
| mmr\_lambda\_0.6 | 24.28% | 7.83% | 4.66% | 24.00% | 7.27% | 5.80% | 12.40% | 0.97% | 1.30% |
| mmr\_lambda\_0.8 | 23.90% | 7.83% | 4.59% | 25.09% | 7.86% | 6.08% | 12.30% | 0.97% | 1.30% |
| mmr\_lambda\_1.0 | 23.03% | 7.16% | 4.51% | 24.77% | 7.90% | 6.07% | 12.07% | 0.97% | 1.27% |

#### Observations

* The MMR-based method doesn’t improve the performance. I think the problem is the cosine similarity doesn’t capture the semantic similarity. The lexicon overlap is small in our data. For example, the cosine for “what we will learn in this class” and “the main topic of the course” are 0.
* However, it does give me the intuition that considering the duplication might improve the performance. For example, when lamda = 0.6 or 0.8, it performs better than 0 or 1.0.

## Adjective Pair

Instead of considering the NP, I will consider only the adjective pair as same as (Yatani et al., 2011).

The POS tagger in Senna is based on the Penn Treebank part-of-speech tags, see

<https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html>

Based on the tag set:

* Definition of adjective: JJ, JJR, JJS
* Definition of noun: NN, NNS, NNP, NNPS
* Step to extract adjective-noun pair
  + Identify the noun and adjective first based on the POS defined above
  + For each noun, locate the nearest adjective (with in the same sentence, either before or after the noun).

There are two variations:

* No Single Noun: If there is no adjective, extraction nothing (This is the method used in (Yatani et al., 2011))
* With Single Noun: However, there are not many adjective words, so if there is no adjective, just the noun is extracted.

In addition, I also test this model under two conditions: (see report 08-07-2014 for detail)

* Hand Weighting: counting with exactly match
* Soft Weighting: counting with shared stem word

### Examples

For Week1, Muddiest Point

TA’s summary:’

1) Grading process [14]

2) Homework assignments [5]

3) Differences between types of bonding

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Hard \* NoSingleNoun |  | Hard \* WithSingleNoun |  | Soft \* NoSingleNoun |  | Soft \* WithSingleNoun |  |
| pre- test | 2 | grading | 7 | confusing grading | 20 | grading | 28 |
| molten tin | 2 | scale | 3 | clear everything | 10 | confusing grading | 27 |
| clear everything | 2 | nothing | 3 | equal grading | 9 | equal grading | 15 |
| confusing grading | 2 | system | 2 | confusing material | 8 | clear nothing | 11 |
| clear today | 1 | pre- test | 2 | clear nothing | 8 | confusing nothing | 11 |
| confusing material | 1 | molten tin | 2 | confusing bit | 8 | confusing material | 10 |
| equal students | 1 | clear everything | 2 | confusing nothing | 8 | clear everything | 10 |
| un part | 1 | hw | 2 | confusing kind | 8 | normalized grades | 9 |
| 3d shape | 1 | bonding | 2 | confusing need | 7 | interesting grade | 8 |
| last semester | 1 | confusing grading | 2 | confusing topics | 7 | confusing bit | 8 |
| clear mud | 1 | clear today | 1 | clear mud | 6 | confusing kind | 8 |
| '" drawing | 1 | confusing material | 1 | clear exercises | 6 | nothing | 8 |
| equal sounds | 1 | equal students | 1 | clear today | 5 | different material | 7 |
| '" wire | 1 | un part | 1 | equal students | 5 | bonding | 7 |
| '" difference | 1 | process | 1 | equal sounds | 5 | confusing need | 7 |
| sure homework | 1 | 3d shape | 1 | pre- test | 5 | confusing topics | 7 |
| different material | 1 | last semester | 1 | different material | 5 | clear today | 6 |
| clear exercises | 1 | clear mud | 1 | normalized grades | 5 | clear mud | 6 |
| normalized grades | 1 | none | 1 | equal method | 5 | pre- test | 6 |
| few bonds | 1 | equal sounds | 1 | equal fairness | 5 | clear exercises | 6 |

### Result of shallow summarization



### Observations:

* With Single Noun improves the performance a lot (the adjective-noun pair for produce review domain cannot be directly used in open survey summarization)
* The soft version improves the performance a lot.
* The AdjNoun model is not better than the NP model.

## Pyramid Evaluation

I’m going to annotate the summary based on the tool DUCView. <http://www1.cs.columbia.edu/~ani/DUC2005/DUCView/>

However, as shown in the introduction, for Pyramid Evaluation, it usually has multiple human summaries. The recommend number of human summary is 4 or 5.

## NP Clustering

TODO

## Paper I read

[1] Radev, Dragomir R., et al. "Centroid-based summarization of multiple documents." *Information Processing & Management* 40.6 (2004): 919-938. [<http://nb.mit.edu/f/15746>]

[2] Carbonell, Jaime, and Jade Goldstein. "The use of MMR, diversity-based reranking for reordering documents and producing summaries." *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1998. [<http://nb.mit.edu/f/15766>]

[3] Rus, V., Lintean, M., Banjade, R., Niraula, N., and Stefanescu, D. (2013). SEMILAR: The Semantic Similarity Toolkit. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, August 4-9, 2013, Sofia, Bulgaria.

## Reference

Yatani, Koji, et al. "Review spotlight: a user interface for summarizing user-generated reviews using adjective-noun word pairs." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011. [<http://nb.mit.edu/f/15423>]