# Summary Report (10/09/2014)

## Overview

In this week, I have done the following things:

* A new algorithm for clustering based summarization
* Malformed Keyphrase filtering

## New algorithm for clustering based summarization

In the previous report, the duplication removing is based on clustering (refer to the report 08-07-2014) by the first two rules as shown below.

* Rule 1: Select only one phrase from one cluster
* Rule 2: The order of the phrases in one cluster is determined by their frequency
* ? Rule 3 (not considered): The order of the cluster is determined by the number of phrases in the cluster. (It is not perfect because it doesn’t consider the frequency of the phrases in the cluster.)

Now, I include the rule 3 and exclude rule 2 by changing the shallow-based summarization algorithm as follows:

Input: A list of documents (students’ responses), D

Output: Top N key phrases that represent the documents, with the limitation that the total number of words is less or equal to N.

Clustering based algorithm:

1. Select the key phrases with certain method for each sentence s in every document
   1. Chunk-based
   2. Syntax-based
   3. Other method [1]
2. Get a collection of all the extracted key phrases as
3. Cluster the phrases with certain method (K-Metroid for example): (how to select K)
   1. K-Means
   2. K-Metroid
4. Rank the clusters by the number of phrases in the cluster (Rule 3)
5. Select one phrase for each of cluster according to the ranking until reaching the N limit (Rule 1)
   1. Select the centroid of the cluster
   2. Select the most frequent one (Rule 2)

Note: the sub bullets (a, b, c) are the possible choices we can try.

The difference of this algorithm and the old one is that the frequency is directly considered by the clustering step (all phrases are passed to the cluster step without compute the Soft-Frequency like before)

### Results

Beyond the five semantic similarity introduced by SEMILAR (i.e. 'greedyComparerWNLin', 'optimumComparerLSATasa','optimumComparerWNLin', 'dependencyComparerWnLeskTanim', 'lexicalOverlapComparer')

I also added the “soft frequency” as the similarity (the total number of shared stemmed words).

However, the similarity value is no longer in [0, 1]. Therefore, I used the formula below to change the similarity (S) to distance (D).

I also run the old method again to get the best performance as a comparison.

Below is the detail results with different setting of K and similarity metrics (some rows are deleted to save the space).



### Best performances with the optimal parameters:



### Observations:

* The new clustering based summarization has a huge win (table above)
  + The best performance of the method (“BestK+Method\_NewClustering”) got the best ones among all the evaluation scores except R2, R-SU4 for MP and R2 for LP. Even if it is not the best one, the performance is close to the best.
* It is too optimistic to compare the previous models with a combination of best values with different parameters. Thus, I picked two real models: one with lambda = 0.6 and “soft frequency” as the similarity metric (“K0.6-npsoft-NewClustering”) ; the other is lambda=0.3 and similarity metric = “optimumComparerLSATasa” (in future, the parameter lambda will be picked up with a development set or cross validation)
  + They outperformance “NP-Soft” on all the metrics (No model did it before) and outperformance “Unigram” except “R1” for LP.

## Malformed Keyphrases

Malformed keyphrases [2] are candidate phrases starting or ending or both in stop words. However, this definition are not suitable for our case, such as the phrase starting with “the” (the different properties, the class), or “a normalized grading scale”.

In our case, we define malformed keyphrase as candidate phrases that are single stop words or starding with a punctuations, such as “nothing, none, we, i”.

I tested this idea with the new algorithm (similarity metric=soft frequency and optimumComparerLSATasa because they perform better than other metrics).

### Results



* By removing malformed keyphrases, the performances become better in general
  + At the same time, the best parameter decreases because the number of candidates decreases.

## Keyphrase

The shallow summarization task is really like the task “keyphrase extraction”. (Question: are they the same?)

I used the state-of-art keyphrase extraction toolkit [Maui](http://code.google.com/p/maui-indexer/), used in [2]. Maui implements a two-stage algorithm for performing the task automatically.

The first stage, candidate generation, identifies candidate topics in a given document. Candidates are either mappings from phrases to terms in the vocabulary (thesaurus in term assignment or Wikipedia in controlled keyphrase extraction), or document phrases (in tagging).

The second stage, filtering, analyzes the properties, or features, of the candidate topics and filters out the most significant ones.

It is a supervised method (training with data set SemEval2010, scientific papers) with the following features:

* Frequency statistics, such as term frequency, inverse document frequency, TFxIDF;
* Occurrence positions in the document text, e.g. beginning and end, spread of occurrences;
* Keyphraseness, computed based on topics assigned previously in the training data, or particular behaviour of terms in Wikipedia corpus;
* Semantic relatedness, computed using semantic relations encoded in provided thesauri, if applicable, or using statistics from the Wikipedia corpus;

In addition, I changed two parameters for our dataset

* The minimal frequency, from 2 to 1 as suggested (because our document is short)
* The output phrase number, from 10 to 20 (to be consistence with the 30 word limitation)

In the post processor, I discards the phrases if the total number of words in the selected phrases are more than 30.

### Results



### Observation

* Directly apply keyphrase extraction method doesn’t get a good performance at least for two reasons
  + The training dataset for the supervised model doesn’t match with our data set
  + The Maui doesn’t consider the similarity between the phrases in the second stage (The output phrases might be similar with each other therefore it reduces the coverage)

### Examples

Below are two sample summary outputs for week1 with Maui and the clustering model.

TA’s summary:

|  |  |  |
| --- | --- | --- |
| **Point of Interest** | **Muddiest Point** | **Learning Point** |
| 1) Group activity of analyzing bicycle's parts [12]  2) Materials processing [6] 3) The main topic of this course [4] | 1) Grading process [14] 2) Homework assignments [5] 3) Differences between types of bonding | 1) Working in a small group [13] 2) Picures and visuals [3] 3) Teaching to others [2] |

|  |  |  |
| --- | --- | --- |
| Maui | | |
| POI | MP | LP |
| hip replacement | normalized score | group interaction |
| learn this semester | bit more discussion | risks in learning |
| replacement in notes | homework is due | taking risks |
| notes Extrusion Different | due The class | taking my own |
| Extrusion Different type | structure regarding assignments | notes I like |
| type of materials | class structure regarding | help |
| uses Most interesting | little vague | group help |
| interesting Class | grading | like group discussions |
| properties and applications | review bonds Normalizing | learning help |
| characteristics of materials | extrusion and wire | group |
|  | need to review | exactly I'm supposed |
|  |  | interacting I like |
|  |  |  |
|  |  |  |
|  |  |  |
| KMedoidMalformedKeyphrase\_0.1\_optimumComparerLSATasa | | |
| POI | MP | LP |
| this class | the test scores | group interaction |
| materials | the grading | notes |
| the grade normalizing | bonds | my learning |
| a bike | the class | activity |
| the group activity | molten tin | this class |
| the bicycle | the grading system | pictures and teamwork |
| the class |  | the objective |
|  |  |  |
|  |  |  |

## Paper I read

[1] Hasan, K. S., & Ng, V. (2014). Automatic Keyphrase Extraction: A Survey of the State of the Art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1262–1273). Baltimore, Maryland: Association for Computational Linguistics. Retrieved from <http://www.aclweb.org/anthology/P14-1119>

<http://www.mendeley.com/share/document/invite/7042f4e465/?utm_medium=email&utm_source=transactional&utm_campaign=share%2Finvitation-document>

## References

[2] Marujo, L., Viveiros, M., & Neto, J. P. da S. (2013). Keyphrase Cloud Generation of Broadcast News. *arXiv Preprint arXiv:1306.4606*. Information Retrieval. Retrieved from <http://arxiv.org/abs/1306.4606>