# Summary Report (10/30/2014)

## Overview

In this week, I have done the following things:

* Summarization based on Candidate Phrase Selection

## Results based on phrase candidate

### Two methods are proposed to select candidate phrase

The candidate phrase selection can be viewed as a segmentation problem. The input is a sentence and a dictionary of possible phrases. The output is the candidate phrases which should be in the given dictionary.

This is just same as the word segmentation for Asia languages. The method to solve this problem is called forward maximal matching (FMM) or backward maximal matching (BMM). It is a greedy algorithm, which scans the input sentence from left to right (FMM) or right to left (BMM) and select the longest possible ngram that is in the given dictionary and then go on.

Usually, BMM is slight better than FMM because it is less disambiguate when using backward. However, in our data set, the output of the two algorithms are exactly the same (strict constraint). The reason is that there are not many candidate phrases selected.

#### Strict Constraint

The ngram should match the phrase in the dictionary exactly. This is too strict. So the stemming constraint is proposed as below.

#### Stemming Constraint

The stemmed ngram should match the stemmed phrase in the dictionary.

### Ranking based on Soft-Frequency

These just are the NP-Soft model but with different candidate phrase selection method.

* NP-Soft: Chunk-based NP selection
* SyntaxNP-Soft: Syntax-based
* CandidateNP-Soft:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | POI | | | MP | | | LP | | |
|  | R1 | R2 | R-SU4 | R1 | R2 | R-SU4 | R1 | R2 | R-SU4 |
| NP-Soft | 35.83% | 8.74% | 9.95% | **34.11%** | 6.92% | 8.74% | 22.43% | 1.93% | 4.00% |
| SyntaxNP-Soft | **37.63%** | 9.98% | **11.74%** | 32.30% | 10.03% | 8.17% | 22.05% | 2.11% | 3.50% |
| CandidateNP-Soft | 22.34% | 7.53% | 4.99% | 25.01% | 8.92% | 6.24% | 8.16% | 0.00% | 0.54% |
| CandidateNP-Stemming-Soft | 24.89% | 7.84% | 6.07% | 27.15% | 9.77% | 6.39% | 10.13% | 0.00% | 1.89% |

### Clustering based

There are two possible similarity metric: npsoft and optimumComparerLSATasa (short for LSA); four possible phrase extraction methods: chunk, syntax, candidate, and candidatestemming.

Thus, there is a combination of 2 \* 4 = 8 possible way for summarization.

#### Observations:

* The syntax-LSA gets the best performance (lambda = 0.1, R2=0.138), which is much better than the chunk-based LSA (lambda=0.4, R2=0.107)
* Generally, syntax method based NP extraction is better than chunk-based method
* The phrase candidate selections doesn’t gain better performance in term of R2. However, I believe the readability would be better.
* Stemming constraint based Candidate Selection is better than strict constraint.

### Difference between four candidate phrase selection methods

The tread of R2 when lambda changes for these method are very different. I’d like to know more about it.

Thus, I perform an analysis about the extracted phrases with the four methods, including

* Ratio phrase/ sen: average number of phrases per sentence
* Total phrase: total number of phrases extracted per lecture per reflect point
* # word/phrase: average number of words per phrases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| type | method | ratio phrase/sen | total phrase | # word/phrase |
| POI | Strict Constraint | 0.436 | 18.583 | 1.574 |
| POI | Stemming Constraint | 0.584 | 24.917 | 1.561 |
| POI | Chunk | 2.594 | 108.417 | 1.956 |
| POI | Syntax | 1.764 | 73.75 | 3.892 |
|  |  |  |  |  |
| MP | Strict Constraint | 0.52 | 23.417 | 1.553 |
| MP | Stemming Constraint | 0.651 | 29.167 | 1.62 |
| MP | Chunk | 2.437 | 109.417 | 2.001 |
| MP | Syntax | 1.632 | 73.333 | 3.957 |
|  |  |  |  |  |
| LP | Strict Constraint | 0.105 | 4.333 | 1.254 |
| LP | Stemming Constraint | 0.254 | 10.417 | 1.117 |
| LP | Chunk | 2.829 | 114.75 | 1.66 |
| LP | Syntax | 2.21 | 89.333 | 2.805 |

### Observations

* The Chunk based method extracted the most number of phrases and they are short (less than 2 words)
* The phrases extracted with the Syntax method are longest (about 3-4 words)
* The phrase candidate methods produces least number of phrases and they are also short. That’s why lambda increases, it is better.

### Examples

Lecture 3

|  |  |
| --- | --- |
| Human Summary | 1) Graphs of attraction/ repulsive & atomic separation [10]  2) Properties and equations with bond strength [7]  3) Coefficient of thermal expansion  4) Activity part III [4] |
| Unigram | graphs, thermal, expansion, bond, understand, strength, class, repulsive, activity, confusing, coefficient, equations, elastic, modulus, graph, attraction, didn, hard, determine, part, iii, hooke, read, bit, small, related, understood, lecture, couldn, worksheet |
| Chunk -LSA-0.4 | coefficient, graphs, the class, thermal expansion, elastic modulus, bond strength, the least bit, part iii, the activity, the concepts, the text, attraction and repulsion, hooke, several slides, property |
| Syntax-LSA-0.1 | the coefficient of thermal expansion relationship to bond strength, the graph, part iii on worksheet in class , comparing metals ., the activity, the graphs of attraction and repulsion |
| candidatestemming-npsort-0.8 | property related to bond strength, coefficient of thermal expansion, metal, elastic modulus, hooke ' s law, activity, atomic structure, heated, energy, activities, bond strength, metals |

This is really a good example to illustrate the benefit with syntax-LSA.

## More suitable baselines

### PhraseMead

The input for Mead are sentences. To adapt Mead to support phrase, I extracted the phrases and feed them into Mead with assuming

* Each phrase is a sentence
* Phrases come from the sentence will be organized as a program

#### Results

Note: all of them has filter out malformed phrases.



Observation:

* PhraseMead is not better than the clustering based method

### Lex-Rank

TODO

## Different clustering methods

TODO

## Quality Prediction

TODO

## Paper I read

Loza, V., Lahiri, S., Mihalcea, R., & Lai, P. (2014). Building a Dataset for Summarization and Keyword Extraction from Emails. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*. Reykjavik, Iceland: European Language Resources Association.

Coursey, K. (2008). Automatic keyword extraction for learning object repositories. *Proceedings of the American Society for Information Science and Technology*, *45*, 1–10.

<http://www.mendeley.com/share/document/invite/1a62a3e9bc/?utm_medium=email&utm_source=transactional&utm_campaign=share%2Finvitation-document>

## Talk I attended

<http://colloquium.lti.cs.cmu.edu/20141024.html>

Title: Exploring the Relationship between Summarization and Diversity

Presenter: [Bruce Croft](http://ciir.cs.umass.edu/personnel/croft.html)

Summary:

General speaking, this is a very good talk that is highly related to the current project.

The basic idea is to combine NLP (summarization) and Information Retrieval (IR) (diversity). Here are his conclusions:

* Diversity and sum keep coming up together. Ranking diversification improving a property of a result summary
* Use summarization can improve diversity
* Better summarization means better diversification

Useful information for this talk:

* There are standard metrics to evaluate diversity in IR: NRBP, alpha-NDCG, ERR-IA, MAP-IA
* The state-of-the-art methods to consider redundancy beside MMR, xQUAD (minimize redundancy: topics), PM-2 (maximize diversity)
* It also proposed a method to extract key phrases based on DSPapprox Algorithm (Lawrie & croft, 2003). However, it is just based on bigram, not real phrase.

## Suggestions from the Wine & Research

* Why only NP is considered?
* Now, for clustering methods, the sentences are treated as equal important. However, multiple sentences coming from same student might be different with one sentence from a student.
* Other clustering method: K-modes [1], K-phototype. (they are designed for non-numeric data)
* Why tf-idf phrase doesn’t work?

[1] Chaturvedi, Anil, Paul E. Green, and J. Douglas Caroll. "K-modes clustering."*Journal of Classification* 18.1 (2001): 35-55.