A "Diamond-Standard" for Evaluating Extractive Summaries and Keyphrases using Abstractive Reference(s)

ABSTRACT

Automatic Summarization and Keyphrase Extraction are two widely used areas in Natural Language Processing and Information Retrieval. A reliable dataset is extremely helpful to the systems performing these tasks. To develop "gold-standards" manually on a large scale dataset is time-consuming and expensive, as well as subjective - differences between summaries (or keyphrases) written by different people can be significant.

Our contribution, in this work, is two-fold: firstly, we provide a dataset with a highly trustable reference set for summaries and keywords, and secondly, we provide an evaluation mechanism for systems that are mostly extractive whereas the references are by nature abstractive.

In case of scientific articles, author assigned abstracts and key-phrases can provide a high quality reference for summarization and key-phrase extraction tasks. We call author generated summary (i.e., abstract) and key-phrases as our *"diamond standard"*. We collect a set of scientific articles, along with author assigned abstracts and keywords, and provide the major sections of the article (excluding abstract, title, and reference list) as a dataset. In order to test the usability of the dataset and the diamond-standard, we use TextRank as a candidate system. As author provided samples can be abstractive by nature, and can vary over length, we compute a relativized score using *i-measure* between the TextRank generated output and the diamond-standard. Systems using our dataset can use the relativized scale (a variant of *i-measure*) to evaluate their performances.

1. INTRODUCTION

Several efforts are going on to create reliably larger datasets for keyphrase extraction and summarization tasks. The Document Understanding Conference (DUC) and the Text Analysis Conference (TAC) are two series of evaluation workshops organized to encourage research in Natural Language Processing and related applications, by providing a large test collection, common evaluation procedures, and a forum for organizations to share their results. TAC comprises sets of tasks known as "tracks," each of which focuses on a particular subproblem of NLP. TAC tracks focus on end-user tasks, but also include component evaluations situated within the context of end-user tasks. The DUC provides the de-facto standard for *unstructured* text data sets which the NLP community uses for evaluating summarization systems. The TIPSTER Text Summarization Evaluation Conference (SUMMAC) published a corpus of 183 documents from the Computation and Language collection. The ACL Anthology Network (AAN) is a manually curated networked database of citations, collaborations, and summaries in the field of Computational Linguistics. A series of workshops on text summarization (WAS 2000-2002), special sessions in ACL, CoLING, SIGIR, and government sponsored evaluation efforts in United States (DUC 2001-DUC2007) have advanced the technology and produced a couple of experimental online systems [8]. However there are no common, convenient, and repeatable evaluation methods that can be easily applied to support system development and comparison among different summarization techniques [4].

Keyphrases for a document concisely describe the document using a small set of phrases (i.e., sequences of contiguous words in a document). Given today's very large collections of documents, these keyphrases are extremely important not only for summarizing a document, but also for the search and retrieval of relevant information. However, keyphrases are not always available directly. Instead, they need to be gleaned from the details in documents. We have found several efforts to accumulate dataset for evaluating keyphrase extraction tasks. Out of several standard ones, [6] is closely related to us. The dataset contains around 200 scientific articles with author assigned and reader assigned keyphrases.

We, in compared to the existing databases, would like to argue that an author is the most suitable person to summarize the article. For example, if a reader's expertise does not fit the domain of the article, the reader-produced summary/keyphrases might not be the best ones to compare with. It is very time consuming and expensive to produce human annotated reference set for a large scale database as well. Hence, we extract author written abstracts and keyphrases to provide a standard reference set, named as *diamond standard*.

Human annotated references are mostly abstractive; whereas

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

system (program) produced ones are extractive. The references and the generated ones might differ due to length as well. Hence, we should move our evaluation techniques from traditional (absolute scaling) methods (precision, recall, f-measure) to a relativized approach, e.g., [2]. In order to show the usage of our dataset we have used a well-known unsupervised approach (TextRank) [5] to produce some candidate output, and we will be providing its' performance statistics using both, the traditional and the relativized scale.

Most of the existing evaluation approaches use absolute scales (e.g., precision, recall, f-measure) to evaluate the performance of the participating systems. *Such measures can be used to compare summarization algorithms, but they do not indicate how significant the improvement of one summarizer over another is* [1]. ROUGE (Recall Oriented Understudy for Gisting Evaluation) [3] is one of the well known techniques to evaluate single/multi-document summaries. ROUGE is closely modelled after BLEU [7], a package for machine translation evaluation. ROUGE includes measures to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans. The measures count the number of overlapping units such as n-gram, word sequences, and word pairs between the machine-generated summary and the reference summaries.

2. RELATIVIZE THE SCALE

While we compare a system generated output with a given reference, we control the *length* of the output so that we get a fair scenario to be compared with. We, then, either find exact string/phrase/word matching techniques or have to employ manual labors for generating paraphrases in order to compare. We propose a modified approach of [2] that not only can adjust to the length variation but also considers WordNet based *IS-A* relationship to define the path similarity between words (synsets).

We will be describing the baseline generation scenario briefly, and then explain how we relate the system generated abstractive/extractive approaches with the provided references.

2.1 A Sound Baseline for All

We consider each document (and generated summary/ key-phrases, given references, etc.) as a *set* of words (or phrases). We are flexible towards the set sizes except for the fact that the size of system generated summary/key-phrases and provided references are shorter than the original document size. At first, in order to draw a base-case scenario, we assume that *references and system outputs are complete subsets of the original document.* Considering these we can relate the baseline to equation 1.

Baseline

Given a set N of size n, and two randomly selected subsets $K \subseteq N$ and $L \subseteq N$ with sizes k and l (say, $k \leq l$), the average or expected size of the intersection ($|K \cap L|$) is

$$avg(n,k,l)_{random} = \frac{\sum_{i=0}^{k} i\binom{k}{i}\binom{n-k}{l-i}}{\sum_{i=0}^{k}\binom{k}{i}\binom{n-k}{l-i}}. \quad (1)$$

For each possible size $i = \{0..k\}$ of an intersecting subset, the numerator sums the product of i and the number of different possible subsets of size i, giving the total number of elements in all possible intersecting subsets. The denominator simply counts the number of possible subsets, so that the fraction itself gives the expected (average) number of elements between two randomly selected subsets.

Eq. 1 is expressed as a combinatorial construction, but the probabilistic one is perhaps simpler: the probability of any element x being present in both subset K and subset L is the probability that x is contained in the intersection of those two sets $I = L \cap K$.

$$\begin{aligned} \Pr(x \in K) \cdot \Pr(x \in L) &= \Pr(x \in (L \cap K)) \\ &= \Pr(x \in I) \end{aligned} \quad (2)$$

Putting another way, the probability that an element x is in K, L, or I is k/n, l/n and i/n respectively (where i is the number of elements in I). From eq. 2 we deduce,

$$\begin{aligned} (k/n)(l/n) &= i/n \\ i &= \frac{kl}{n} \end{aligned} \quad (3)$$

2.2 The Relativized Scale

A direct comparison of an observed overlap (say, ω), seen as the intersection size of two sets K and L, consisting of lexical units like unigrams or n-grams drawn from a single set N is provided by the *i-measure*:

$$\begin{aligned} \textit{i-measure}(N,K,L) &= \frac{observed\_size\_of\_intersection}{expected\_size\_of\_intersection} \\ &= \frac{|K \cap L|}{\frac{|K| \cdot |L|}{|N|}} = \frac{\omega}{\left(\frac{kl}{n}\right)} = \frac{\omega}{i} \end{aligned} \quad (4)$$

2.3 Connect Relativized scale to Absolute Scale

Recall, Precision, and *f-measure* are the re-known absolute scales to define the performance of a system. Recall (r) is the ratio of *number of relevant information received to the total number of relevant information in the system.* Precision (p), on the other hand, is the ratio of *number of relevant records retrieved to the total number (relevant and irrelevant) of records retrieved.* Assuming the subset with size k as the gold standard, we define recall, and precision for the randomly generated sets as:

$$r = \frac{i}{k} \quad \text{and} \quad p = \frac{i}{l}$$

$$\textit{f-measure} = \frac{2pr}{p+r}$$

f-measure (the balanced harmonic mean of p and r) for these two random sets can be redefined using eqn 3 as:

$$\textit{f-measure}_{expected} = i/((l+k)/2) \quad (5)$$

Let, for a machine generated summary L and a reference summary K, the observed size of intersection, $|K \cap L|$ is ω.

$$r = \frac{|K \cap L|}{|K|} = \frac{\omega}{k} \quad \text{and} \quad p = \frac{|K \cap L|}{|L|} = \frac{\omega}{l}$$

f-measure, in this case, can be defined as,

$$\textit{f-measure}_{observed} = \omega/((k+l)/2) \quad (6)$$

By substituting ω and i using eq.6 and 5, we get,

$$\textit{i-measure}(N,K,L) = \frac{\textit{f-measure}_{observed}}{\textit{f-measure}_{expected}} \quad (7)$$

Interestingly, *i-measure* turned out as a ratio between the observed *f-measure* and the expected/ average *f-measure*.

In other words, *the i-measure is a form of f-measure with some tolerance towards the length of the summaries/words.*

3. FROM ABSTRACTIVE TO EXTRACTIVE

After exploring the existing automatic summarization and keyphrase extraction algorithms, we have discovered that majority of the works focus on *extractive* methods. Most of the reference sets, on the contrary, are abstractive by nature. When a human is appointed to produce a summary (or write a set of representative phrases), he improvises the sentences/phrases as much as he can. Hence, *our hypothesis, $K \subseteq N$ is not always true.*

In this circumstances, we can deviate from the exact string/ phrase matching approach to a *fuzzy/weighted* approach. *Our motive is to relate the basic units (words, in this case) found in the system generated outputs to the units from the reference set.* To satisfy our goal, we need a large ontology, or thesaurus. We, in this work, use WordNet to find hypernym path similarity for evaluation.

WordNet can be seen as a combination of dictionary and thesaurus. Both nouns and verbs are organized into hierarchies, defined by hypernym or *IS-A* relationships. The words at the same level represent synset members. Each set of synonyms has a unique index. We have used the *path similarity* between two word-senses while relating one with the other. While evaluating summaries, we apply a *POS-tagger* to fine-tune the synsets. While evaluating keyphrases, we considered each word as *noun*, and find the path similarity using the *first sense* between two words.

3.1 Scoring Approach

For the scientific article set, we have the diamond standard and we use the TextRank as a test system. For each paper, the TextRank algorithm generates a summary and a set of keywords. The TextRank graph builds a directed edge between the words due to their co-occurrence within a window range and/or the synset based similarity measure determined by WordNet synset relations.

After generating keyphrases and summaries, we use a modified version of eq. 4 to score the system. We adapted the WordNet *hypernym tree* to detect *path similarity* between two synsets of the system output and the diamond standard. The similarity score ranges from 0 to 1. Then eq. 8 is used to find the score.

$$score(L) = i\text{-}measure(N, K, L) \times \sum_{w_l \in L} max\{path\_similarity(w_l, w_k), \forall w_k \in K\} \quad (8)$$

The evaluation code published with this paper considers only one reference (the diamond standard) per document. But it can be easily tuned to handle multiple references using similar approaches as ROUGE or Credibility-based Scoring [2].

4. BUILDING THE DATASET

We have collected data (papers) from several proceedings of WWW and KDD. We avoided posters as their internal structure is even complex and contains more figure than texts. Our dataset is composed with a total of 2000 articles. We prepare the dataset in the following structure:

- There are k folders. Each folder contains the following files:
  - original paper in pdf format,
  - paper in text format,
  - title of the paper,
  - author-written abstract (diamond summary),
  - author provided keyphrases (diamond keyphrases),
  - body of the paper excluding references, and section headers (dataset)
  - textrank generated summary
  - textrank generated keyphrases
  - evaluation statistics (precision, recall, f-measure, i-measure)

5. REFERENCES

[1] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell. Summarizing text documents: Sentence selection and evaluation metrics. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 121–128, New York, NY, USA, 1999. ACM.

[2] F. Hamid, D. Haraburda, and P. Tarau. Evaluating text summarization systems with a fair baseline from multiple reference summaries. In *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings*, pages 351–365, 2016.

[3] C.-Y. Lin. Rouge: a package for automatic evaluation of summaries. pages 25–26, 2004.

[4] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 71–78, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[5] R. Mihalcea and P. Tarau. TextRank: Bringing order into texts. In *Proceedings of EMNLP-04and the 2004 Conference on Empirical Methods in Natural Language Processing*, July 2004.

[6] T. D. Nguyen and M.-Y. Kan. Key phrase extraction in scientific publications. In *Proceeding of International Conference on Asian Digital Libraries*, pages 317–326, 2007.

[7] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[8] D. Radev, S. Blair-Goldensohn, Z. Zhang, and R. Raghavan. Newsinessence: A system for domain-independent, real-time news clustering and multi-document summarization. In *Proceedings of the First International Conference on Human Language Technology Research*, 2001.