https://doi.org/10.2352/issn.2168-3204.2019.1.0.27

Functional Applications of Text Analytics Systems

Steven J Simske[1], Marie Vans[2] ([1]Colorado State University, Fort Collins CO, [2]HP Inc, Fort Collins CO, USA)

Abstract

Text analytics can provide a wide breadth of valuable information, including summarization, clustering, classification, and categorization to enable better functional interaction with the text. This includes improved search, translation, optimization, and learning. In this paper, we describe advanced analytical approaches used to enable improved utility of the text documents and information later. This adds value to the preservation of the information and provides new access points to the information. We emphasize the role of functional approaches to testing and configuration of these systems, with the view that the primary role of archiving is to make the content as re-usable, re-purposeable, and discoverable as possible.

Introduction

Text analytics consist at the most elementary level of the statistics about a text element, which includes the word count, the word histogram, and the word frequency histogram. Most text documents of value are related to other—sometimes many other—documents, and so analytics describing the relative frequency of terms in a document compared to its peers are important for defining key words (tagging, labeling, indexing), search-responsive terms (query terms), and compressed versions of the documents (key words, summary, etc.).

Document text is initially analyzed for part of speech tagging and compound relationships (compound nouns, auxiliary verbs, etc.), since the part of speech of words significantly impacts their utility in downstream analytics such as summarization, clustering, classification, and categorization.

With these statistical analytics in place, we can then proceed with the more functional analytics of search, translation, optimization, and learning. Generation of analytics such as is aided by a hybrid, ensemble, or other combinatorial approach in which two or more effective analytics processes are used simultaneously, and their outputs combined to form a better "consensus". Additional value to the preservation of the information is provided through these methods. Also, since they encompass capabilities of two or more knowledge-generating systems, they can create a "superset" of access points to the data generated. We also describe the role of functional approaches in the testing and configuration of these systems.

Linguistics and NLP

One of the first, and obvious, connections between linguistics and NLP (natural language processing) is the choice of algorithms for the primary NLP tasks (part of speech tagging, categorization, word sense, N-grams, collocations, etc.) based on the language identified [1]. For example, articles are substantially different in comparing English and Japanese.

A functional approach to linguistics and NLP is an iterative refinement algorithm. First the language family can be identified from the character set. Next, the individual language is identified from the word counts. The dialect, if appropriate, can be identified from the rare terms within the language that occur with disproportionately high frequency in the document. This might include regional idiomatic expressions as well as variant spellings. Finally, jargon and slang dictionaries can be used to assign the document to specific specialties, trades, or other subcultures.

Summarization

Summarization is a powerful analytic in its own right, serving as a proxy for the document it compresses. Effective compressing leads to a concept we herein designate *compressive substitution*. By compressive substitution we mean a level of compression that does not result in significant loss of functionality. If the summarized (or otherwise compressed) content functions as well as the original text, then we have a compression value of 1.0 In some cases, such as the idealized curve shown in Figure 1, compressive substitution may exceed the performance of the original content, particularly if we are measuring normalized performance such as the product of accuracy and efficiency. Finding the same content in half the time might give a value of 2.0 for compressive substitution. In some search processes, the accuracy alone may improve because the summarized content provides more germane input than the complete documents, allowing more accurate search.

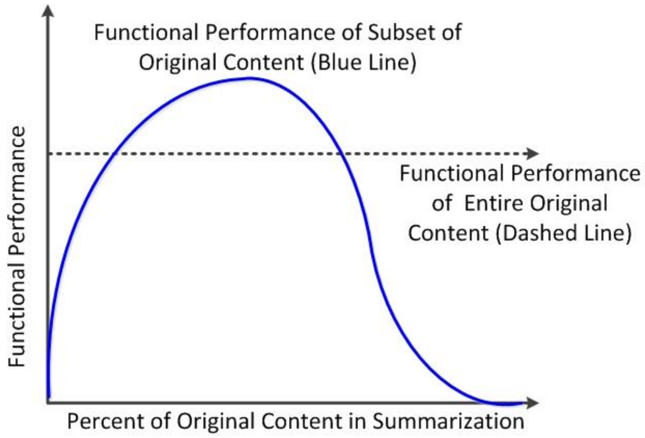

Figure 1. Idealized functional performance curve, where the functional performance peaks at a particular level of summarization. The solid line (blue in original) represents the performance of the summarized content in a functional task (e.g. search accuracy multiplied by search speed). The functional performance axis is usually nonzero at the asymptotic value. Please see text for details.

Thus, compressive substitution can be an absolute (directly comparing the accuracy, precision, recall, throughput, robustness, etc. of summarized and original content) or relative (comparing the product of accuracy, precision, etc., with throughput). Regardless of how we define the compressive substitution, the main point is that we can decide how large a percentage of the original content to include in the summary based on how well the so-summarized content performs in its downstream use cases. We now apply these principles in a few example applications.

Clustering, Classification and Categorization

Clustering, classification, and categorization are sometimes used relatively interchangeably. However, here we consider clustering to correspond to associating content with content very much like it; classification as deciding what type of content is in multiple clusters, often when compared to a pre-defined set of classes; and categorization to be the post-classification step wherein the clustered and classified content is tagged. These tags in turn can be used as an automatically-generated set of search queries, indices, etc.

Summarized content can be used as the primary content for clustering, classification, and categorization. The performance of the system using summarized content can be compared to the performance on the original content, generally resulting in a curve similar to Figure 1, although the asymptotic behavior for absolute measurements will meet the dotted line as the percent of content increases, of course (since it becomes the original content with summarization percentage = 100%). Importantly, different summarization strategies [2] can be employed in addition to different percentages, meaning that we can develop a family of curves of the type shown in Figure 1 in order to functionally optimize our summarization. Note that we might have a different (strategy, percentage) for each function—i.e. a separate one for clustering, classification, categorization, and query set generation.

Not surprisingly, the percentage of each document needed for summarization is a function of overall corpus size (generally positive correlation), breadth of materials in the corpus (generally negative correlation), and type of summarization used (extractive generally works better than semantic summarization for these tasks).

Translation

Summarization and its counterpart query set generation can be used to quantitatively grade the accuracy of multiple translation engines [3]. Here, there is definitely an optimal percentage of summarization that leads to the best overall behavior as determined by *equivalence of query behavior* when comparing the source and translated documents. For this purpose, the summarization is an auxiliary function, not optimized for its own utility but instead used to determine the behavior of the translated queries together with the translated corpus. The best translator is deemed to be the one that provides a translated {document, query set} that behaves the most similarly to the {document, query set} of the original language text.

Optimization

Optimization can be tied to absolute or relative objective functions. Examples already mentioned include optimizing the summarization strategy and percentage to provide optimal:

(1) Clustering
(2) Classification
(3) Categorization
(4) Tagging/indexing
(5) Query set generation
(6) Translation

However, virtually any other text processing task—including those that occur "upstream" of these six, can be performed using summarized representations of the documents. While part of speech tagging and word sense analytics are usually expected to perform better on larger text data sets, it is sometimes possible to get better performance on the difficult-to-analyze portions of a corpus on compressed text content (that is, summaries). This seems contradictory, but the fact that only the most salient text is analyzed means that the NLP algorithms might fight unusual word usage, including non-traditional word sense, better than if the less salient portions of the document "overwhelm" the analytics. That is, slang, jargon, and other neologisms are concentrated in the summaries in comparison to the document as a whole, and so provide a higher density for these linguistic "anomalies" than does the entire document.

Learning

One of the main challenges of producing an archival representation of a corpus is to encourage its reuse. Tagging, indexing, clustering, and classification all provide additional access points for the corpora. From a utility standpoint, however, we would like large corpora to be useful for training and learning purposes. Learning is about the proper sequencing of many documents to enhance understanding and retention by the reader. As such, we would generally like to know which document to provide a reader once she has finished with a specific document. We need to have the right amount of overlap for reinforcement, and the right amount of non-overlap to encourage the ingestion of new concepts and facts.

Summarization is a spatial representation of a corpus. Each document that is summarized can remain so summarized, so long as proper testing and optimization have been employed, and each summary occupies a spot in the overall mapping of the corpus content. Learning, however, is concerned with the *temporal* representation of the corpus. When we have completed a document, what is the next one to read? The answer, no surprise, depends on a number of factors, including but not limited to the following:

(1) Proficiency of the reader
(2) Number of documents in the same cluster
(3) Number of documents in the same class
(4) Number of documents with the same categories
(5) Number of documents the reader has time to read

Given this, it is clear that learning represents a vast opportunity for additional research. In fact, because of the high rate of change in documents themselves (particularly web documents but extending to books and magazines which are almost universally available in streaming form), this is likely to remain the biggest opportunity for text archiving research for some time to come.

Testing and Configuration

We have provided herein the means to use a compressed proxy (summarized) representation of a document in place of the original document set. However, a concern around using such a compressed proxy set is that training, validation, and optimization is based on a quasi-experimental approach, rather than a proper experimental approach. A quasi-experiment is empirical—that is, control and experimental group assignment are *a posteriori*, so that it is used to estimate the causal impact of an experimental factor, but without random assignment. This means such an experiment is almost certain to be impacted by one or more confounding factors.

We use an analogy here to illustrate this concern. There are usually ethical and/or structural exigencies which prevent a proper experimental design in many psychology-related research areas. One example is to test the impact of smoking on the development of another disorder such as lung cancer. It would be unethical to randomly assign participants in the experiment to either the control or one of the experimental (e.g. 0.5, 1, or 2 packs day) groups, since assignment to an experimental group would have considerable health risks (not to mention it may be difficult to enforce!). Thus, for this experiment, the assignments are after the fact, and the experiment designer does not have the ability or choice to change the independent variable. Thus, if only such a study were available, a cigarette manufacturer might argue that there may be a predisposition of people to smoke who already have a higher than average genetic risk of lung cancer. Some might argue that this is statistical apologetics, giving the cigarette manufacturers a loophole. It's not—I'll close that loophole shortly. However, from the standpoint of a quasi-experiment, it can reasonably be argued that a physiological lung defect such as weakened alveolar linings, dilated bronchi, etc., may make a person more likely to smoke: the smoking might constrict the bronchi to normal levels and so alleviate discomfort. The onus should then be on the cigarette manufacturers to establish that constricted bronchi lead to a higher lung cancer rate, but that only delays the creation of another confounding factor to keep the argument going. Fortunately, there are means of establishing a proper experiment *a posteriori*. Here is where identical (that is, monozygotic) twins are an absolute boon to psychological research. Because they are born with equivalent genetic information, if we find identical twins with different smoking behavior, we can act as if they were assigned to these different groups *a priori* (since from the genetic standpoint their assignment is random). This elevates the quasi-experiment to an experiment.

Applying this to document analytics, we need to make sure that the summarization data (which is only quasi-experimental) can substitute for the full-text documents (which, with proper definition of training, validation, and testing sets, provide an experimental data set). Thus, proper equivalency comparisons (e.g. similarity of search query behavior on the original and compressed proxy corpora) are essential.

Sample Application: Synonymic Search

One of the most important text-based analytics is search. Functional measurements of search are relatively straightforward in some cases; for example, the use of search behavior to rank the efficacy of different translation engines, as mentioned above.

Ranking the efficacy of different search engines, however, generally takes human-driven ground truthing. We used a 10-class, 1000-document, human ground-truthed subset of the CNN corpus[4] for a simple test of search accuracy. Search queries based on the key class terms for each of the 10 classes were expanded using 2, 4, 6, 8, 10, 12, 14, 16, 18, or 20 synonyms to augment the class-specific search query (which consisted of the top 10 key terms of each class). For example, a search query for class "Travel" includes the term "Boat" which has synonyms such as "Ship". Using the WordNet NLTK and the wup_similarity (meaning Wu-Palmer Similarity), the similarity between "boat" and "ship" is 0.91. All synonymic terms for each of the terms in the original search query have the Wu-Palmer similarity calculated, and the terms with the top 2, 4, 6, …, 20 weights are added to the search query.

The terms of interest to this sample application include the following:

1. Corpus Size—Sum of all Positives and Negatives—here it is a total of 1000
2. Positives—search results returned to you (matches)
3. Negatives—not returned to you as search results (non-matches)
4. True Positives (TP)—search results you find that are actually from the correct class
5. False Positives (FP)—search results you find that are not actually from the correct class, but are wrongly returned from the search
6. True Negatives (TN)—search results from the incorrect classes that are left out (correctly not returned as matches)
7. False Negatives (FN)—search results from the correct class that are left out (not returned as matches, but should have been returned as matches)
8. Precision (p)—percent of search results that are useful. The value p=TP/(TP+FP).
9. Recall (r)—percent of all useful search results actually returned to you. The value r=TP/(TP+FN).
10. Accuracy (a)—harmonic mean of precision and recall, and the recommended metric for optimization of the search—the value a= 2pr/(p+r).

The latter three values—that is, p, r, and a—are shown in Table 1. Clearly, accuracy peaks in the range of 6-12 added synonyms (that is, expanding the 10-word search query to 16-22 terms). Importantly, adding synonyms above 12 does not further increase accuracy. Also, including less than 6 synonyms results in lower accuracy.

Table 1: Results for the synonymic search experiment

#Synonyms	Precision p	Recall r	Accuracy a
0	0.744	0.390	0.512
2	0.741	0.400	0.519
4	0.783	0.470	0.588
6	0.736	0.530	0.616
8	0.684	0.540	0.603
10	0.648	0.570	0.606
12	0.622	0.610	0.616
14	0.568	0.630	0.597
16	0.485	0.650	0.556
18	0.366	0.640	0.465
20	0.300	0.620	0.404

The specifics of how synonymic search will behave for corpora of different sizes and number of classes is unknown. However, the data shown here indicate that it can be a significant positive influence on document classification accuracy. The peak value, 0.616, is 0.104 higher than the value when using no synonyms, 0.512. This is a 21.3% reduction in error rate.

Discussion and Conclusions

In this paper, we have highlighted some of the exciting ways in which corpora can be enhanced—and optimized—to provide reuse, new access points, and improved behavior in aggregate. This includes the use of large corpora for learning, which is the current area most needing of additional research. One largish corpus, the CNN corpus, was used to illustrate the functional use of synonyms to improve search accuracy.

Acknowledgments

Special thanks to Dr. and Professor Rafael Lins, Universidade de Pernambuco, Recife, Brazil, for his leadership on creating the CNN corpus. This work was sponsored in part by NSF Award 1842577, supported by NSF EAGER: ISN: Advanced Analytics, Intelligence and Processes for Disrupting Operations of Illicit Supply Networks, Program Manager Georgia-Ann Klutke.

References

[1] R. Rehurek, M. Kolkus, “Language Identification on the Web: Extending the Dictionary Method,” computational Linguistics and Intelligent Text Processing, pp. 357-368, 2009.

[2] R. Ferreira, R.D. Lins, L. Cabral, F. Freitas, S.J. Simske, M. Riss, “Automatic Document Classification using Summarization Strategies,” 2015 ACM Symposium on Document Engineering, pp. 69-72, 2015.

[3] S.J. Simske, I.M. Boyko, G. Koutrika, “Multi-engine search and language translation,” EDBT/ICDT Workshops, pp. 188-190, 2014.

[4] R.D. Lins, S.J. Simske, L.S. Cabral, G.F.P. Silva, R. Lima, R.F. Mello, L. Favaro, “A multi-tool scheme for summarizing textual documents,” 11st IADIS international conference WWW and INTERNET 2012, pp. 1-8, 2012.

Author Biographies

Steve Simske received his PhD in electrical engineering from the University of Colorado (1990) to augment a BS (Marquette University) and MS (Rensselaer Polytechnic Institute) in Biomedical Engineering. He was a Research Fellow and Director in HP Labs for the last decade of his time there (1994-2018). He is currently Professor in Systems and Mechanical Engineering at Colorado State University. His research includes imaging, security, anti-counterfeiting, analytics, and sensing. He is currently President of IS&T.

Marie Vans is currently a Research Scientist with Hewlett-Packard Labs in Fort Collins, Colorado. Her main interests are security printing and imaging for document workflows, statistical language processing, and other approaches to document understanding. She holds a Ph.D. in Computer Science from Colorado State University. She also recently completed a second master’s degree in Library and Information Science at San José State University.

.