UNIVERSITY OF WATERLOO

Faculty of Engineering

Department of Management Sciences

MSCI 541 – Project Report

Stefanno Da Silva
ID: 20508389

April 18th, 2016

Introduction

In IR systems users must describe with words the content of the documents in which they are interested in retrieving. The system must then, through the use of complex algorithms, decide on what documents are most likely to contain information pertinent to the user needs. In simple words, a relevant document in the collection should contain the words utilized by the user to describe his information needs (This collection of words is also known as the "query"), and documents containing more instances of those words are more likely to be relevant than documents with less instances or completely lacking the words in the query. The words contained within each documents and query are also known as tokens.

In this project we built two custom IR System to retrieve the top 1000 most relevant documents for 45 different queries/topics by utilizing BM25 scoring as a metric to evaluate the relevancy of each document. The document collection belongs to the LA Times and contains exactly 131,896 different documents, these being articles and news from the most varied topics. All of the 45 topics have specific relevant documents which will be then compared to the relevant documents retrieved by the system. The objective of this report is to analyse the difference in the quality of results of both systems. The first system, also our baseline, implemented Simple Tokenization while the second system made use of Custom Tokenization (CUS). The two IR engines were evaluated on basis of their respective nDCG@1000 scores.

This report assumes that the reader posses a certain background on the topic of Information Retrieval, and will focus on describing the procedures and methodologies implemented during the development of our custom IR system. Specific terms will be briefly described.

Methodology

Materials

The materials utilized during the course of this project were a list of 45 topics and a document collection belonging the LA Times containing multiple news reports. The 45 queries utilized were pre-determined for the project, and each is formed by a title that briefly describes the topic of interest (See *Appendix B: List of Search Topics*). The 131,896 documents of the collection were gzip'd into a file and organized on the SGML (Standard Generalized Markup Language) format, where the documents are uniquely identified by the tag <*DOCNO*>. The documents are divided in several segments, each under certain tags, as seem in *Appendix A: LA Times Sample Document*. When reading a document, we should only tokenize words found in between specific tags, these tags being <*HEADLINE*></*HEADLINE*> and <*TEXT*></*TEXT*> and <*GRAPHIC*> </*GRAPHIC*> and <*SUBJECT*></*SUBJECT*>.

Tokenization

One of the most important aspect of Information Retrieval is the process of tokenizing the words of the query and the document, allowing the system to build an index of all the words contained within the collection, as well as all of the documents that include each word. During this project two different methods for tokenizing text were utilized: Simple Tokenization and Custom Tokenization (CUS).

Simple Tokenization: In simple tokenization a string of text is put into lowercase then split on whitespace and any special character such as "-", "%" and "@" is striped from the string. Each formed words is considered an individual token. As an example, *"NF-k B/CD28-responsive"* is divided into the following tokens: *"nf", "k", "b", "cd28"* and *"responsive"*. The system utilizing Simple Tokenization will be our baseline run.

Custom Tokenization: The main objective of this project was to measure the efficiency of CUS over the original tokenization methods implemented. The custom tokenizer (CUS) was a tokenization model presented by D. Trieschnigg, W. Kraaij and F. de Jong for their paper on "The Influence of Basic Tokenization on Biomedical Document Retrieval". CUS combines compound tokenization, use of stop words and Porter Stemming. For this experiment the encoding for the Porter Stemmer utilized was developed by Brad Patton. For this project, CUS was modified to

better adapt to the document collection and the search engine, thus ignoring special Greek characters such as Kappa (κ) or Gamma (γ).

Initially the text to be tokenized is transformed into a sequence of lowercase characters and split on whitespace. Each word is then stripped of any special character while numbers and letters are separated. As an example, the word "*B/CD28-responsive*" is split into the tokens *"b", "cd", "28"* and *"responsive"*. For each token we verify if they are contained within a list of 418 stop words, and, if they are the token is thrown away and will not be added to the inverted index nor the compounded token. Each token is then stemmed and then united once more to generate the compounded token. At the end of the process "*B/CD28-responsive*" is divided into 5 tokens: *"b", "cd", "28"*, *"responsive"* and *"bcd28respons".*

The Inverted Index

All modern search engines make use of the *inverted index* as a method to organize information regarding each token. It is called inverted index because we usually think of a document as a collection of words, and by "inverting" this concept, we must think of documents as entities associated to specific words. The IR systems developed during the course of this project makes use of In-Memory inversion (Which means that the inverted index is stored in RAM instead of being Disk-Based). The reasoning behind this decision that due to the relative small size of the collection (While 131,896 documents might appear to be large, companies such as google and Microsoft work with collections containing billions of documents.), storing the index in memory will not cause any performance issues. The inverted index was built using a dictionary of lists, in where the key represents a token and each item on a list contains within it a document number and a count (The count represents the number of times the token has appeared on the document). A representation of the inverted index can be seen in *Figure 1 - Inverted Index Representation*.

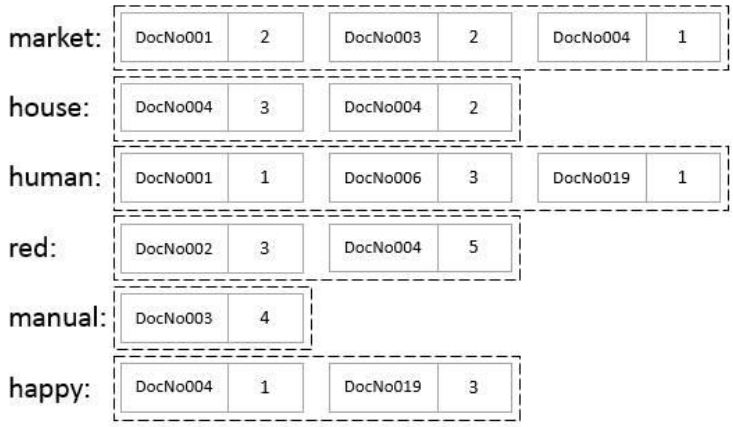


Figure 1 - Inverted Index Representation

For each document, all of the relevant text is tokenized and then loaded into an inverted index. The methodology applied to build the inverted index is described as following: A temporary index is built for each document. This index contains a list of all the tokens, and the count of times they appear on that documents. Once the temporary index is built we evaluate each entry and verify if the token is already present in the inverted index. If not present, the token is added, together with the current document number and its respective count for that one token. If already present, the document and the count are added to the list containing all other documents associated to that token.

The BM25 Ranking Algorithm

The algorithm utilized to score and evaluate the relevancy of each document was the BM25, one of the most effective and popular rankings algorithm. We judge the relevancy of each document based on the sum of the scores of all terms present in the query in respect to their presence, or not, in the document. The following form of the formula was implemented for this project:

$$\sum_{i \in Q} \log \frac{N - ni + 0.5}{ni + 0.5} * \frac{(k1+1) * fi}{K + fi} * \frac{(k2+1) * qfi}{k2 + qfi} \text{ and } K = k1 * \left[(1 - b) + b * \frac{dl}{avdl}\right]$$

In where N represents the number of documents, *ni* the number of documents containing term *i*, *fi* the frequency of term *i* in the document, *qfi* is the frequency of the term within the query and the variables *dl/avdl* stand each for document length and average document length. The parameters k1, k2 and b are set to the values of 1.2, 7 and 0.75 respectively.

Performing the Search

For each topic we first perform a similar tokenization method as utilized when building the index. Once all the tokens are determined we retrieved each token respective index entry that contains the list with all their relevant documents. From that point on we iterate through each token and their documents, utilizing BM25 to calculate each document respective score for one term. For documents appearing in multiple tokens, the scores obtained for each token are added, thus forming a final relevancy score. Each document and its relevancy score is added to a priority queue that organizes the documents from most relevant to least relevant. Finally, for each topic we output into a text file the first 1000 entries of the priority queue, representing the 1000 most relevant documents.

Analysing the Results

A python program was developed to analyse the results obtained from each IR engine. For each topic the program iterates through all positions of the ranking, and checks in each position if the retrieved document is part of the relevant collection for that topic. Information regarding all the relevant documents is found in a previously provided *qrels* file. The nDCG scores for each topic are calculate and outputted into a text file. This is repeated for both the results obtained from the baseline run and the implementation of CUS. In Excel we import both text files and analyse the results by calculating the mean difference between each system nDCG and by performing a paired t-test.

Results

The results show that the system making use of Simple Tokenization obtained an average nDCG of 0.429 while the system implementing CUS achieved an average nDCG of 0.494 (See *Figure 2 - Baseline (Simple Tokenization) VS. CUS Tokenizer Results*). Compared to the baseline that utilizes Simple Tokenization we can see an improvement of 15.09% when CUS is implemented to the search engine. To further verify if the difference between the two systems is in fact significant, a paired t-test between the means was performed. The paired t-test results in a two tail p-value of 0.000182 (See *Figure 4 - Paired t-Test Results*), which show us that the difference of 0.0648 is unlikely to occurs by purely chance, thus concluding that Custom Tokenization does in fact produce better results. For graphical representation of the results please see *Appendix C: Additional Figures*.

	Baseline nDCG	CUS Tokenizer nDCG
Topic 401	0.2786	0.3760
Topic 402	0.3072	0.6127
Topic 403	0.7574	0.7523
Topic 404	0.1519	0.2022
Topic 405	0.1724	0.1628
Topic 406	0.6168	0.7570
Topic 407	0.5425	0.5060
Topic 408	0.4667	0.4558
Topic 409	0.2891	0.3010
Topic 410	1.0000	1.0000
Topic 411	0.3726	0.5720
Topic 412	0.7615	0.8300
Topic 413	0.1496	0.2560
Topic 414	0.3247	0.3439
Topic 415	0.3904	0.3904
Topic 417	0.7232	0.7569
Topic 418	0.3361	0.5946
Topic 419	0.6940	0.7877
Topic 420	0.8937	0.8622
Topic 421	0.2557	0.2883
Topic 422	0.6791	0.6834
Topic 424	0.2331	0.5830
Topic 425	0.6942	0.8566
Topic 426	0.1823	0.1836
Topic 427	0.3040	0.3806

	Baseline nDCG	CUS Tokenizer nDCG
Topic 428	0.3386	0.3281
Topic 429	0.5840	0.8838
Topic 430	0.6823	0.7499
Topic 431	0.3295	0.6735
Topic 432	0.1160	0.0817
Topic 433	0.1101	0.1061
Topic 434	0.4315	0.4252
Topic 435	0.3370	0.2711
Topic 436	0.3969	0.4060
Topic 438	0.4034	0.4629
Topic 439	0.1887	0.1896
Topic 440	0.8386	0.8338
Topic 441	0.6984	0.8497
Topic 442	0.2121	0.2217
Topic 443	0.4298	0.5770
Topic 445	0.4018	0.4373
Topic 446	0.2073	0.2340
Topic 448	0.2351	0.2155
Topic 449	0.0943	0.0883
Topic 450	0.6981	0.6941

Average Baseline nDCG:	0.4291
Average CUS Tokenizer nDCG:	0.4939

Figure 2 - Baseline (Simple Tokenization) VS. CUS Tokenizer Results

	Baseline nDCG	*CUS Tokenizer nDCG*
Mean	**0.4291**	**0.4939**
Variance	**0.0564**	**0.0664**
Observations	**45.000000**	**45.000000**
Pearson Correlation	**0.911018**	
Hypothesized Mean Difference	**0.000000**	
df	**44.000000**	
t Stat	**-4.087893**	
P(T<=t) one-tail	**0.000091**	
t Critical one-tail	**1.680230**	
P(T<=t) two-tail	**0.000182**	
t Critical two-tail	**2.015368**	

Figure 3 - Paired t-Test Results

Discussion

Problems with Simple Tokenization

Simple Tokenization seems to fails to consider how users are likely to behave in reality, thus producing lacking results. When implementing Simple Tokenization, we are merely interested in striping the text of all non-alphanumeric characters. This system fails to consider two things: First is that many times the user, when performing a search, might be also interested in finding results that contain variations of the words utilized in the query. For example, when searching for "Winners of the 2012 summer Olympics", it is very likely the document containing the phrase "United Stated ends up winning the summer 2012 Olympics" will be of interest to the user, even through "winners" and "winning" are different words. Second, the words "the" and "of" aren't of much interest to the users, they were merely added to the phrase for grammatical reasons, however are still considered when calculating the relevancy scores.

The issues above end up creating a IR system that is "dumb" to variation, and can only consider the specific words utilized in the query, even when in reality, documents always utilize variations of the same word to express the same idea. For example, in the query above the system will likely rank an article about the fashion store "Winners" higher and than article talking about the winners of the summer Olympics, as the later will utilize variations of the word "winners" such as "win", "winning", "winner", and other, that while not necessarily match the token "winner", would be considerate relevant by a reader. Failing to disconsider common words such as "the" and "of", highly frequent words and make up most of the vocabulary of a document, also cause problems when calculating the BM25 scores, as the system becomes highly dependant on smoothing the values to decrease the gap between large an short document. However, even by smoothing the terms, large and irrelevant documents will still obtain a few extra points due to the large repetition of those words, possibly being ranked higher than another actually relevant but shorter document.

Improving the Solution with Custom Tokenization

Custom Tokenization is an attempt to consider the issues previously ignored by Simple Tokenization. Two aspects of CUS are assumed to contribute to better performance of the search engine: Addition of a list of 418 stop words that are to be ignored the tokenizer, and the implementation of the Porter Stemmer, which aims to stem each word into their roots, allowing for variations of the same word to be considered as the same token, thus giving room for more

flexibility when evaluating a document. The results in fact show a better performance when CUS is implemented, however we want to evaluate which aspect was the one to contribute most to the difference. In order to evaluate that, two additional runs of the engine were performed: The first run removes the list of stop words, and those will be included in the index. The second run removes Porter Stemming from the tokenization process, while keeping the list of stop words. After running both variations of the engine, the following results were obtained:

	Stop Words Only Run	Porter Stemmer Only Run
Topic 401	0.2660	0.3637
Topic 402	0.3071	0.6136
Topic 403	0.7523	0.7574
Topic 404	0.1528	0.2018
Topic 405	0.2023	0.1555
Topic 406	0.7370	0.6611
Topic 407	0.5429	0.5073
Topic 408	0.4624	0.5005
Topic 409	0.2891	0.2891
Topic 410	1.0000	1.0000
Topic 411	0.3842	0.5701
Topic 412	0.7625	0.8303
Topic 413	0.1505	0.2702
Topic 414	0.3244	0.3240
Topic 415	0.3904	0.3904
Topic 417	0.7403	0.7300
Topic 418	0.3298	0.6140
Topic 419	0.6934	0.8048
Topic 420	0.8904	0.8619
Topic 421	0.2570	0.2889
Topic 422	0.6804	0.6795
Topic 424	0.2440	0.5790
Topic 425	0.7020	0.8515
Topic 426	0.1942	0.1821
Topic 427	0.3076	0.3812

	Stop Words Only Run	Porter Stemmer Only Run
Topic 428	0.3393	0.3299
Topic 429	0.8768	0.8712
Topic 430	0.6821	0.7499
Topic 431	0.3313	0.6806
Topic 432	0.1309	0.0676
Topic 433	0.1067	0.1088
Topic 434	0.4281	0.4281
Topic 435	0.3283	0.2747
Topic 436	0.3974	0.4107
Topic 438	0.4056	0.4633
Topic 439	0.1897	0.1880
Topic 440	0.8388	0.8335
Topic 441	0.7424	0.8541
Topic 442	0.2271	0.1968
Topic 443	0.4808	0.5931
Topic 445	0.4373	0.4632
Topic 446	0.2074	0.2343
Topic 448	0.2313	0.2169
Topic 449	0.0934	0.0892
Topic 450	0.6965	0.6966

Stop Words Only Run nDCG:	**0.4430**
Porter Stemmer Only Run nDCG:	**0.4924**

Figure 4 - Stop Words Only Run VS. Porter Stemmer Only Run

The results show a that Porter Stemming clearly has a greater impact on the final results. The run that ignored stop words only shows a very small improvement over the baseline, which was expected, as BM25 takes into consideration differences in document lengths. To properly evaluate the impact of adding stop words into the IR system further studies should be conducted. Porter Stemming however show a much greater difference between the baseline run and the run that implemented Porter Stemming only. A reason for such difference is that the Porter Stemmer allows for variation between words, which is more realistic approach when evaluating a document. The results obtained with the Porter Stemmer shows the importance the of this recall enhancing technique, as more relevant documents are retrieved and added higher to the rankings.

Conclusion

The objective of this project was to evaluate how different tokenization methods can affect the quality of the retrieved results. The first method implemented was Simple Tokenization, which was the same model utilized Homework 1, while the second method utilized a more complex and complete tokenization method, the Custom Tokenization. Simple Tokenization is a very limited approach on tokenization, as it limits the flexibility of the engine in evaluating words of the same root, and fails to ignored commonly repeated words such as “the” and “of”, which can cause a small, but noticeable change in the recall of the documents. Custom tokenization addresses those problem by implementing a list of stop words that are to be ignored, as well as implementing Porter Stemming, a technique that allow us to classify words on the roots. Custom Tokenization, mostly due to the addition of the Porter Stemmer, produce higher quality results, with an significant improvement of 15.09% over the baseline.

References

[1] Trieschnigg, Dolf, Wessel Kraaij, and Franciska De Jong. "The Influence of Basic Tokenization on Biomedical Document Retrieval." Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '07 (2007): Web. 24 Mar. 2016.

[2] Smucker, Mark D. "Chapter 9: Information Representation." Interactive Information Seeking, Behaviour and Retrieval. By Ian Ruthven and Diane Kelly. London: Facet, 2011.

[3] Smucker, Mark D., James Allan, and Ben Carterette. "A Comparison of Statistical Significance Tests for Information Retrieval Evaluation." Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management - CIKM '07 (2007): Web. 24 Mar. 2016.

[4] Croft, W. Bruce., Donald Metzler, and Trevor Strohman. Search Engines: Information Retrieval in Practice. Boston: Addison-Wesley, 2010. Print.

[5] McCaffrey, James. "Priority Queues with C#." Visual Studio Magazine. N.p., 11 Feb. 2012. Web. 24 Mar. 2016.

Appendix A: LA Times Sample Document

<DOC>
<DOCNO> LA010189-0001 </DOCNO>
<DOCID> 1 </DOCID>
<DATE>
<P>
January 1, 1989, Sunday, Home Edition
</P>
</DATE>
<SECTION>
<P>
Book Review; Page 1; Book Review Desk
</P>
</SECTION>
<LENGTH>
<P>
1206 words
</P>
</LENGTH>
<HEADLINE>
<P>
NEW FALLOUT FROM CHERNOBYL;
</P>
<P>
THE SOCIAL IMPACT OF THE CHERNOBYL DISASTER BY DAVID R. MARPLES (ST. MARTIN'S PRESS: $35, CLOTH; $14.95, PAPER; 316 PP., ILLUSTRATED; 0-312-02432-0)
</P>
</HEADLINE>
<BYLINE>
<P>
By James E. Oberg , Oberg, a space engineer in Houston, is the author of Uncovering Soviet Disasters: Exploring the Limits of Glasnost (Random House).
</P>
</BYLINE>
<TEXT>
<P>
The onset of the new Gorbachev policy of glasnost, commonly mistranslated as openness but closer in connotation to candor or publicizing, has complicated the task of Soviet secret-keepers and has allowed substantial new Western insights into Soviet society. David R. Marples' new book, his second on the Chernobyl accident of April 26, 1986, is a shining example of the best type of
non-Soviet analysis into topics that only recently were absolutely taboo in Moscow official circles.
</P>
<P>
The author, a British-educated historian and economist, is a research associate
with the Canadian Institute of Ukrainian Studies at the University of Alberta,
and the academic style of the book is undisguised. However, its intended audience is the general public, and anyone interested in nuclear power, or Soviet economy and society, or human drama, or just plain sleuthing state secrets, will find hitherto unpublished revelations and explanations of the event and its continuing aftermath.

</P>
<P>
The effects of Chernobyl reverberated throughout so many facets of Soviet society that a continuous coherent narrative is probably impossible. Marples discusses half a dozen major themes arranged in a fairly arbitrary order (as indicated by the frequent and helpful cross references throughout the text) and
succeeds in mapping out his main themes. The personal interests of each reader
determine which of the sections may be deemed too detailed and which too sketchy, but considering the need for such a comprehensive overview, the levels
are generally appropriate.
</P>
<P>
The book is, on the one hand, not a light read, and an executive summary might
have been possible in a quarter the length. But, on the other, so many of the judgments depend on a subtle interpretation of a multitude of sources that the
author is obligated to present the raw data for the reader's inspection. The modular nature of the book also allows a reader to skip, browse, and revisit earlier sections, aided by a convenient internal organization and a thorough index.
</P>
<P>
First in the world's attention, and in the text, is a discussion of the human victims of the accident. The official tally is 31 (only about 20 names have ever been released), but Marples suspects there were other short-term radiation
victims. A large number of unnecessary late-term abortions were also performed
on local women, and by rights those unborn babies count as casualties. Widespread "radiophobia" led to restricted diets which created malnourishment and subsequent disease in thousands of people. The tens of thousands of people
taking part in cleanup operations were never included in official totals of those exposed. Since the book went to press, Soviet military sources have referred to at least one death in the actual reactor entombment program.
</P>
<P>
But the greatest toll is likely to occur with the delayed deaths. Here, Marples
encounters for the first time the soon familiar theme of official Soviet myth-making around the event: Reality is twisted to serve state policy objectives, which include calming an alarmed public with assurances that all is
well when it isn't.
</P>
<P>
And thus is born what he properly labels the "myth of Chernobyl," the official
line that the disaster provided a test that Soviet society passed with honor. "In the Soviet view," he writes, "it was first and foremost a victory, a story
with an ending, and an ending that was triumphant."
</P>

<P>
Thus, when sober Western medical estimates placed the future "excess cancer deaths" at several tens of thousands, both in the Soviet Union and in Europe (a few tenths of a percent elevation of the natural cancer rate), the Soviets reacted furiously. The estimates are branded "nonsense" and the estimators are dismissed as "panic mongers" promulgating "anti-Soviet venom."
</P>
<P>
Subsequently the author addresses themes of environmental impact, economic and political repercussions, public images, and the recovery operations. Along the way, Marples provides a damning list of examples in which Soviet officials attempted to retreat behind old-style cover-ups and outright lies. False information was issued on radiation levels, on subsequent accidents at the site, on contamination levels of the Kiev water supply, on severe discipline against non-volunteer cleanup personnel, on reactor entombment schedules and on operator training levels.
</P>
<P>
A severe 1986-1987 countrywide electrical power shortage was officially denied although it was real enough to compel the restart of three Chernobyl reactors in explicit violation of Soviet safety regulations. Design deficiencies of the Chernobyl-style reactors were downplayed and human errors were declared to be the primary culprit.
</P>
<P>
Ultimately, observes the author, "It is ironic that in an era of openness, Chernobyl may have been both the pioneer of glasnost under Gorbachev and then subsequently its first casualty." He ultimately concludes, "Aspects of the disaster . . . have rarely been dealt with thoroughly or even honestly by Soviet sources." Hence the need for this book, a need which is admirably fulfilled despite the many remaining mysteries and uncertainties.
</P>
<P>
The July, 1987, trial of reactor personnel marked a full circle of disclosure.
Journalists were allowed into the pre-scripted first and last days, but the weeklong deliberative sessions were held in secret and no word of their substance has ever been released.
</P>
<P>
The propaganda purpose of the trial and surrounding official publicity, he maintains, had one goal: "To divert culpability from the party hierarchy, in Kiev and especially in Moscow." This is precisely the theme I have also encountered in my own investigations of aerospace accidents of the past. Where individual human failings led to catastrophe, a sanitized story may eventually be released, but where Kremlin policy led to disaster (such as the Nedelin catastrophe of 1960 or the Soyuz-1 disaster in 1967), the entire event remains

absolutely off limits to glasnost.
</P>
<P>
The closing blow-by-blow description of the nuclear power debate presages a dramatic event which occurred too recently for inclusion in this first edition.
Viktor Legasov, tagged by the author as one of the country's two leading pro-nuclear advocates, actually was sinking into private despair over the poor
implementation of safety standards. In the end, he made his final and most eloquent testimony to this despair on the second anniversary of the accident, by committing suicide. For several weeks the Soviets tried to sit on the circumstances of his "tragic death," even issuing official non-explanations which asserted that the death was not due to medical effects of radiation. Finally, crusading journalist Vladimir Gubarev, with access to Legasov's notebooks, broke the story in Pravda. Readers of this book will come to know these and other characters so well that the suicide fits right into the "big picture" of the catastrophe's social impacts.
</P>
<P>
For an author to so accurately describe a social milieu that subsequent unpredictable events only enhance his insights is testimony to the highest quality of scholarship. Readers of Marples' book will rarely be surprised as the Chernobyl catastrophe's consequences continue to unfold in the future.
</P>
</TEXT>
<GRAPHIC>
<P>
Photo, Chernobyl Then and Now :Photographs of the damaged reactor taken before
the construction of its concrete "sarcophagus" are, for obvious reasons, aerial
photographs. Left, an artist's reconstruction of the reactor as it would have looked from the ground before the sarcophagus was in place. The point of view is the same as that of an official Soviet photograph, right, taken as the entombment neared completion.
</P>
</GRAPHIC>
<TYPE>
<P>
Book Review; Main Story
</P>
</TYPE>
</DOC>
<DOC>

Appendix B: List of Search Topics

Topic 401: foreign minorities, Germany

Topic 402: behavioral genetics

Topic 403: osteoporosis

Topic 404: Ireland, peace talks

Topic 405: cosmic events

Topic 406: Parkinson's disease

Topic 407: poaching, wildlife preserves

Topic 408: tropical storms

Topic 409: legal, Pan Am, 103

Topic 410: Schengen agreement

Topic 411: salvaging, shipwreck, treasure

Topic 412: airport security

Topic 413: steel production

Topic 414: Cuba, sugar, exports

Topic 415: drugs, Golden Triangle

Topic 417: creativity

Topic 418: quilts, income

Topic 419: recycle, automobile tires

Topic 420: carbon monoxide poisoning

Topic 421: industrial waste disposal

Topic 422: art, stolen, forged

Topic 424: suicides

Topic 425: counterfeiting money

Topic 426: law enforcement, dogs

Topic 427: UV damage, eyes

Topic 428: declining birth rates

Topic 429: Legionnaires' disease

Topic 430: killer bee attacks

Topic 431: robotic technology

Topic 432: profiling, motorists, police

Topic 433: Greek, philosophy, stoicism

Topic 434: Estonia, economy

Topic 435: curbing population growth

Topic 436: railway accidents

Topic 438: tourism, increase

Topic 439: inventions, scientific discoveries

Topic 440: child labor

Topic 441: Lyme disease

Topic 442: heroic acts

Topic 443: U.S., investment, Africa

Topic 445: women clergy

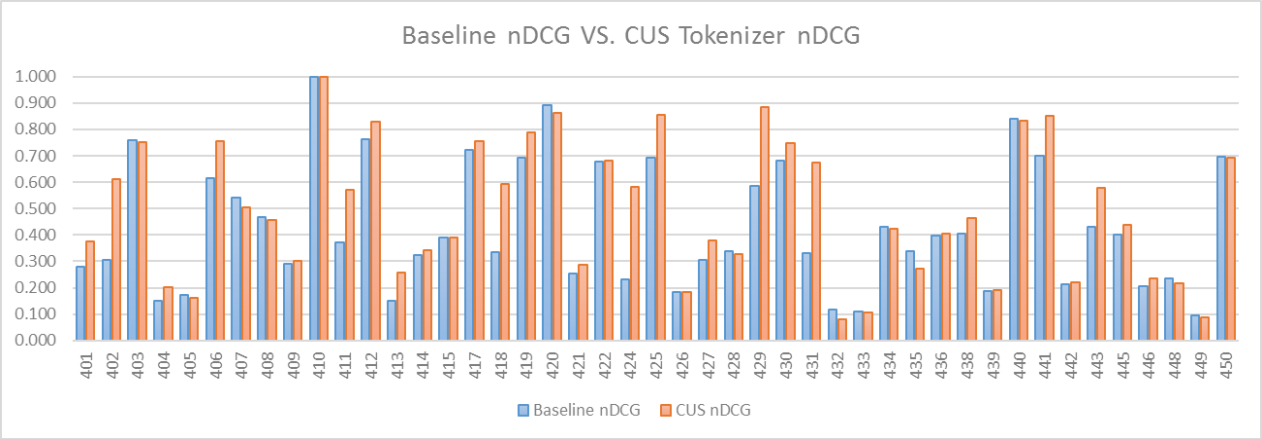
Topic 446: tourists, violence

Topic 448: ship losses

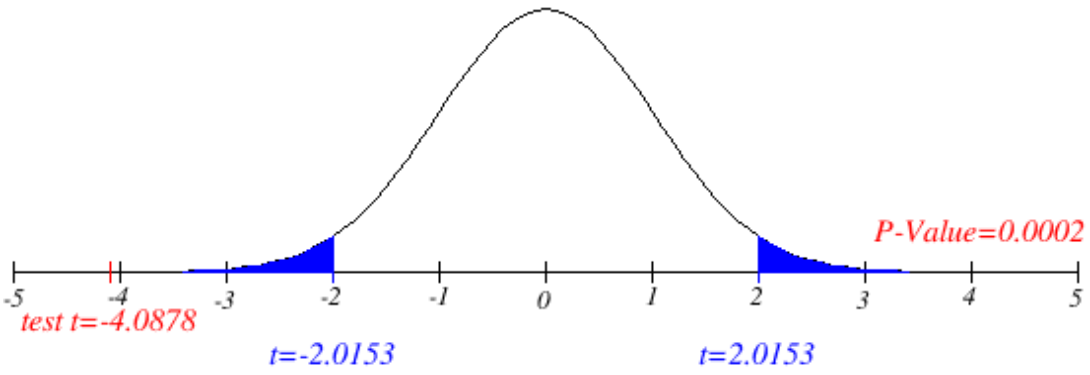
Topic 449: antibiotics ineffectiveness

Topic 450: King Hussein, peace

Appendix C: Additional Figures



Baseline nDCG VS. CUS Tokenizer nDCG



t-Test Curve Graph