2015 IEEE 14th International Conference on Machine Learning and Applications

Summary Sentence Classification using Stylometry

Rushdi Shams
Department of Computer Science
The University of Western Ontario
London, Ontario, Canada N6A 5B7
Email: rshams@alumni.uwo.ca

Robert E. Mercer
Department of Computer Science
The University of Western Ontario
London, Ontario, Canada N6A 5B7
Email: mercer@csd.uwo.ca

***Abstract*—Summary sentence classification is an important step to generate document surrogates known as summary extracts. The quality of an extract depends much on the correctness of this step. We aim to classify potential summary sentences using a statistical learning method that models sentences according to a linguistic technique which examines writing styles, known as *Stylometry*. The sentences in documents are represented using a novel set of stylometric attributes. For learning, an innovative two-stage classification is set up that comprises two learners in subsequent steps: K-Nearest Neighbour and Naïve Bayes. We train and test the learners with the newswire documents collected from two benchmark datasets, viz., the CAST and the DUC2002 datasets. Extensive experimentation strongly suggests that our method has outstanding performance for the single document summarization task. However, its performance is mixed for classifying summary sentences from multiple documents. Finally, comparisons show that our method performs significantly better than most of the popular extractive summarization methods.**

***Keywords*—Summarization, classification, machine learning, data mining, natural language processing, text mining, stylometry.**

I. Introduction

One of the challenges to lessen the effect of today's *information overload* [1] is producing summaries automatically that are short in length, concise in nature, and rich in language properties [2]. In general, summaries are produced either from a single document or from multiple documents. *Extracts* are summaries that are constructed by using *copy and paste* of text units such as clauses, sentences, and paragraphs from source texts [3]. On the other hand, *abstracts* are summaries that are re-generated by relating these text units using textual cohesion and entailment. Overall, an extract is useful to understand the main idea of a source text while its abstract is better for conceptualizing the subject matter [4]. Empirically, the summarizers that simply extract salient sentences from source texts perform better for large-scale applications and therefore receive much of the attention [5]. Keeping this in mind, we put the focus of this paper on extracts—and not on abstracts—that are generated from both single and multiple documents.

Summary sentences are sentences chosen as text units to be included in extracts [3]. The classification of summary sentences from source texts can be seen as a binary classification problem in statistical learning. Over the last decade, an array of methods have been proposed on this topic and on a good note, many are regarded as the domain's state-of-the-art [5]. In their study, Lloret and Palomar [6] showed that to classify summary sentences a linguistic theory named the *Code Quantity Principle* [7] can be considered. According to this principle, the codification of important information in text by humans, so that this information gets more attention, is a process that combines human cognition, psychology, and language. The principle states that the most important information within a text contains more lexical units such as syllables, words, and phrases [8]. These units, and many others, are also the key elements of *stylometry*—a method to model writing styles in texts. Summary sentences are more important than non-summary sentences. Therefore, modelling them using stylometry should result in a good classification.

In this paper, we report on a method to classify summary sentences using stylometry. We model the sentences of a set of documents using 87 stylometric attributes and use these attributes as features for a two-stage classification technique comprising the K-Nearest Neighbour and Naïve Bayes classifiers. The documents are collected from two benchmark datasets for single and multi-document text summarization named the CAST datasets [9] and the DUC2002 collection [10]. Results show that our method has outstanding performance for single document summarization; the performance for multi-document summarization is somewhat mixed. Comparisons show that the proposed method yields much better results than most of the popular summarization methods.

The next section describes work related to this research. Section III discusses the datasets used and outlines the methods we followed. Results of this work can be found in Section IV. Finally, Section V draws conclusion to this paper.

II. Related Work

Many studies have reported summary sentence classification methods. We limit our discussions here to those that have used the same summarization data as ours.

Berker and Güngör [11] used lexical chains to represent lexical cohesion in a document. The chains were then weighted and used as machine-learning attributes to rank and order its sentences. Overall, the method worked remarkably well on the CAST dataset [9] when compared to classical summarization attributes. Mitkov *et al.* [12] investigated the effect of using coreference resolution for single document summarization. Their reported performances on the CAST dataset indicate that this does not bring in any significant benefit. In their extensive study, Villatoro-Tello *et al.* [13] argued that most frequency-based attributes (e.g., sentence position, word frequency, or cue words) are domain-specific. Instead of these well-known attributes, they modelled sentences using an *n*-gram model. The results were extremely competitive on the CAST dataset.

978-1-5090-0287-0/15 $31.00 © 2015 IEEE
DOI 10.1109/ICMLA.2015.181

Dataset	Total Sentences	Summary Sentences (+)	Regular Sentences (−)	Imbalance Ratio
CAST-15	2,579	315	2,264	7.2
CAST-30	2,579	667	1,912	2.9
DUC-200	20,800	491	20,309	41.4
DUC-400	20,800	956	19,844	20.8

TABLE I: Summary of the datasets used in this experiment.

Leskovec *et al.* [14] represented documents as semantic graphs, which is a combination of the logical forms: the subject-predicate-object (*SPO*) representation of its sentences. The method generated two semantic graphs: one for the original document and one for its summary produced by human annotators. A Support Vector Machine (*SVM*) classifier was trained on the semantic graphs to identify the *SPO* triples and was evaluated on test documents. Both the CAST and the DUC [10] datasets were used for training and testing. The outcomes of this study showed a remarkable improvement. Later, Leskovec *et al.* [15] re-generated the semantic graphs using the linguistic structures around the noun and verb phrases as additional components to the *SPO* triples. These additional components further improved their results. Ferreira *et al.* [16] worked on multi-document summarization. They proposed a three-stage algorithm that converts a text into a graph model, identifies salient sentences from the graph using Text Rank [17], and groups sentences based on their similarities. On the DUC dataset, they outperformed all the systems that participated in the DUC2002 competition. The Columbia summarizer [18], a well-known operational text summarizer, has been a regular participant in the Document Understanding Conferences (DUC) and is recognized as one of the state-of-the-art summary tools. The summarizer has two major components: *MultiGen* [19] to generate summaries from single-event documents and *Dissimilarity Engine for Multidocument Summarization* [20] to generate summaries from documents on multiple events. It has been evaluated on the DUC summary sentence extraction tasks. On average, its performance was in the top three among the 10 submitted systems in the DUC2002 competition.

III. Materials and Methods

A. Summarization Data

We have used four benchmark datasets in this experiment. For single document summarization, we have selected the CAST dataset [9], a subset of the Reuters Corpus [21] containing 147 newswire articles. Approximately 15% of the document sentences are marked as *essential* (CAST-15) and an additional 15% as *important* (CAST-30) for the summary. From this collection, we have selected the set of 89 documents that had a single annotator called *Annotator-1* and have modelled the sentences in the CAST-15 and CAST-30, separately. Out of 2,579 sentences, the CAST-15 and CAST-30 datasets contain 315 and 667 sentences labeled as *summary sentences*, respectively (details can be found in Table I).

For multi-document summarization, we have selected the highly regarded dataset collections prepared for the Document Understanding Conference 2002, widely known as the DUC2002 collection [10]. We are particularly interested in this data because the 2002 conference was the last one that proposed a contest to generate extractive summaries [16] and our primary focus is the classification of sentences for extract generation. All other related subsequent DUC[1] and Text Analysis Conference (TAC) shared tasks[2] have "abstracts" and not "extracts" as their gold standard reference summaries. In this collection, one of the datasets contains 59 clusters of 533 newswire articles on 30 different topics. The articles in each cluster belong to the same topic. Each cluster has a 200-word (DUC-200) and a 400-word (DUC-400) summary interpreted as *extracted summaries* since the sentences in the summaries are extracted directly from the articles in the cluster. Out of 20,800 sentences, the DUC-200 dataset has 491 sentences and the DUC-400 dataset has 956 sentences labeled as *summary sentences*. The imbalance ratio (Regular Sentences:Summary Sentences) of these two datasets is much higher than that of the CAST datasets (see Table I).

B. Descriptions of Attributes

The intuition that stylometric attributes can be useful to indicate text importance or informativeness is backed up by a linguistic theory named the *Code Quantity Principle* [7] that states that authors deliberately or subconsciously change their writing styles for the sentences that are more informative. Since the principle is yet to be explored in the summarization domain, we have extended the idea of this principle by creating a set of features that represent the writing variations of an individual author of news articles. We have represented each sentence in the datasets as $(\vec{x}, y)$, where $\vec{x} \in \mathbb{R}^{87}$ is a vector of the 87 stylometric attributes and $y \in \{+, -\}$ is the sentence label. A sentence has the label $+$, if it is a summary sentence, $-$ otherwise. The labels are provided by the human annotators of the CAST and DUC datasets. Details of the stylometric attributes are provided in the following discussion.

1) Text Complexity Attributes: In this category, we have selected 28 attributes that contribute to the complexity of text (see Table II, A1–28). Every sentence has a level of reading complexity and many *de facto* standard scores such as Fog Index, Smog Index, Flesch Reading Ease Score, Forcast, and Flesch-Kincaid Index are used to measure it. Calculation of these scores is mostly based on the use of simple words (words with at most two syllables) and complex words (words with more than two syllables) in a text unit. In this study, all of the aforementioned text complexity scores and the frequency of simple and complex words are used as attributes. The Fog Index measures the proportion of complex words in a text unit. We have created the *Simple Word Fog Index* attribute (the relative use of simple words in a sentence). Another such modified attribute is the *Inverse Fog Index*, the arithmetic inverse of the Fog Index. The details of the readability scores are beyond the scope of this paper and can be found elsewhere [22]. The rest of the attributes in this category are self explanatory: the average word length is the average number of syllables present in the words of a sentence and the number of syllables is simply the frequency of syllables in a sentence. The boolean attributes—long and short sentence—distinguish sentences with more than 20 words.

2) Word-level Attributes: This category contains 52 attributes (see Table II, A29–80), each calculated by considering

[1] See: http://duc.nist.gov/

[2] See: http://www.nist.gov/tac/

No.	Description	No.	Description
A1	Fog Index	A45	Numeric† %
A2	Fog Index†	A46	Lowercase
A3	Simple Word Fog Index	A47	Lowercase %
A4	Simple Word Fog Index†	A48	Lowercase†
A5	Inverse Fog Index	A49	Lowercase† %
A6	Inverse Fog Index†	A50	Uppercase
A7	FORCAST	A51	Uppercase %
A8	FORCAST†	A52	Uppercase†
A9	SMOG Index	A53	Uppercase† %
A10	SMOG Index†	A54	Long Word
A11	FKRI	A55	Long Word %
A12	FKRI†	A56	Long Word†
A13	Flesch	A57	Long Word† %
A14	Flesch†	A58	Unique Word
A15	Complex Word	A59	Unique Word %
A16	Complex Word %	A60	Unique Word†
A17	Complex Word†	A61	Unique Word† %
A18	Complex Word† %	A62	Repeated Word
A19	Simple Word	A63	Repeated Word %
A20	Simple Word %	A64	Repeated Word†
A21	Simple Word†	A65	Repeated Word† %
A22	Simple Word† %	A66	Conjunction
A23	Word Length	A67	Number
A24	Word Length†	A68	Determiner
A25	Syllable Count	A69	Preposition
A26	Syllable Count†	A70	Adjective
A27	Long Sentence	A71	Noun
A28	Short Sentence	A72	Pronoun
A29	Word Count	A73	Adverb
A30	Function Word	A74	Verb
A31	Function Word %	A75	Interjection
A32	Content Word	A76	Foreign
A33	Content Word %	A77	List
A34	Alphanumeric	A78	Possessive
A35	Alphanumeric %	A79	Particle
A36	Alphanumeric†	A80	Symbol
A37	Alphanumeric† %	A81	Character
A38	Alphabetic	A82	Character†
A39	Alphabetic %	A83	Character per Word
A40	Alphabetic†	A84	Character per Word†
A41	Alphabetic† %	A85	Character without Space
A42	Numeric	A86	Character without Space†
A43	Numeric %	A87	Special Character
A44	Numeric†	A88, A89	Positive/Negative Distribution

TABLE II: The list of stylometric attributes used in this study: text complexity attributes (A1–A28), word-level attributes (A29–A80), and character-level attributes (A81–A87). The attributes A88 and A89 are the positive and negative distributions provided by the K-Nearest Neighbor classifier and used in the second stage of our two-stage classification approach. Attributes with † symbol means that they are calculated by removing the function words.

a sentence as a bag-of-words. A standard English *function word lexicon* is used to distinguish function and content words. Long word is the percentage of words with more than five characters. An *alpha-numeric* word must contain both letters and digits. The Stanford part-of-speech tagger [23] is used to tag each word's part-of-speech (details can be found in [24]) and considered the tags as attributes (see Table II, A66–80).

3) Character-level Attributes: Seven character-level attributes (see Table II, A81–87) includes attributes such as total characters including and excluding whitespaces, characters per word, and special characters.

Note that we have calculated the attribute values with and without function words. The exceptions are the long/short sentence, word frequency, function word, content word, parts-of-speech, and special character attributes which are calculated without removing the function words.

4) Attribute Importance: Determining salient attributes to decide the class of data instances is highly recommended in machine learning methods [25] [26]. Therefore, to observe the importance of the stylometric attributes, we have used a well-known algorithm named `Boruta`[25] that exploits Random Forest learners to iteratively remove attributes which are proven to be less important than random probes according to a statistical test. The selection process is both unbiased and stable, and its usefulness has been successfully demonstrated on an artificial dataset. We have applied `Boruta` on all of our datasets and the results are illustrated on the next page in Figure 1 (for the attribute numbers in the figure, refer to Table II). In the figure, the `x-axis` represents the attributes and the `y-axis` refers to their importance. In addition, *important* attributes are represented in *green*, *unimportant* attributes in *red*, and the attributes that are neither important nor unimportant are in *yellow*. Surprisingly, the results demonstrate the strength of the character-level attributes. For all four datasets, most attributes from this category have the highest *importance*. The only exception is the *special character* attribute: it has some importance for the `DUC` datasets, but is unimportant for the `CAST` datasets. Interestingly, part-of-speech attributes like *interjection, foreign*, and *list* are unimportant for all the datasets since none of the datasets contain words from these categories. There are other attributes that are unimportant across all the datasets: *long sentence*, percentage of *unique words* without function words, and frequency and percentage of *repeated words* without function words. It was surprising to us since we had expected at least *long sentence* to be a strong attribute because of the length constraint put on the summary extracts during the dataset curation. Attributes related to *alpha-numerics, numerics, uppercase, adverbs*, and *symbols* are found to be unimportant for the `CAST` datasets. In contrast, *nouns* are among the most important attributes for the `DUC` datasets and *verbs* are found to be strong for the `CAST-30` dataset.

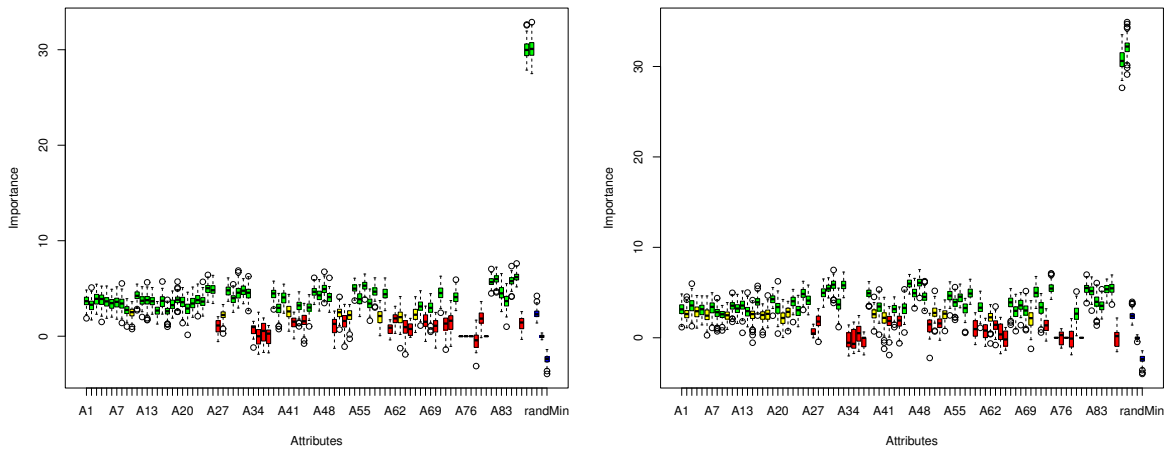
The software package to generate the stylometric models of the sentences[3] and the data files supporting the results of this paper[4] are publicly available. Details regarding the attributes A88 and A89 can be found in Section III-C.

C. Classifier Design

Multi-stage or cascading learning [27][28] is a special case of *ensemble learning*. As the name suggests, several classifiers $C_1, \ldots, C_{n-1}, C_n$ are staged serially so that C_{i+1} learns not only from the attributes of the training instances but also from the class distributions of these instances provided by C_i (e.g., if C_i is a decision tree, the class distributions are probability values of class membership for the training instances). Multi-stage learners are often as fast and as good as ensemble learners but only require simple learners in their cascades.

To classify summary sentences, we have designed a two-stage learner: our first stage involves a K-Nearest Neighbour learner [29] while our second stage is a Naïve Bayes learner

[3] http://cogenglab.csd.uwo.ca/tools/text-summarization-tool-2014.zip
[4] http://cogenglab.csd.uwo.ca/datasets/cast-duc2002-datafiles-2014.zip



(a) Importance of attributes for the `CAST-15` dataset.

(b) Importance of attributes for the `CAST-30` dataset.

(c) Importance of attributes for the `DUC-200` dataset.

(d) Importance of attributes for the `DUC-400` dataset.

Fig. 1: Attribute importance for the datasets using the `Boruta` algorithm.

[30]. There are several reasons for choosing these learners for our two-stage classification. First, both learners are *stable*: a small change in the training data rarely affects performance. Second, we are interested in exploiting the strengths of both discriminative (K-Nearest Neighbour) and generative (Naïve Bayes) learners. Third, both learners are simple as their objective functions are based on probabilities. Last but not least, both learners—especially Naïve Bayes—perform well in text-based classification. We have used the implementations and the default parameter values of these learners found in the Weka machine learning toolkit 3.7 [31].

The overall learning is evaluated using a stratified 10-fold cross validation. Treating each dataset independently, the values for the 87 stylometric attributes are computed for each sentence. Then, each dataset is randomly divided into 10 equal-sized stratified sets. Stratification means that the $+$ and $-$ classes in each set are represented in approximately the same proportion as in the full dataset. One set is used for evaluation and the remaining sets are used for construction of a K-Nearest Neighbour classifier (stage 1 of 2). This classifier then generates two probability values (one for each class) for each instance in the evaluation set. This cross-validation process is then repeated until each of the 10 sets is used exactly once as the validation data for a K-Nearest Neighbour classifier. Now, in addition to the 87 stylometric attributes, each instance in the dataset has been assigned two more attributes. So, each instance is now represented by 89 attributes in addition to the human-assigned *class* attribute. Using these attributes, a Naïve Bayes classifier then generates models in a stratified 10-fold cross validation (stage 2 of 2). We report the average values for the 10 folds of the measures described in Section III-D.

Finding a good value of K for a K-Nearest Neighbour

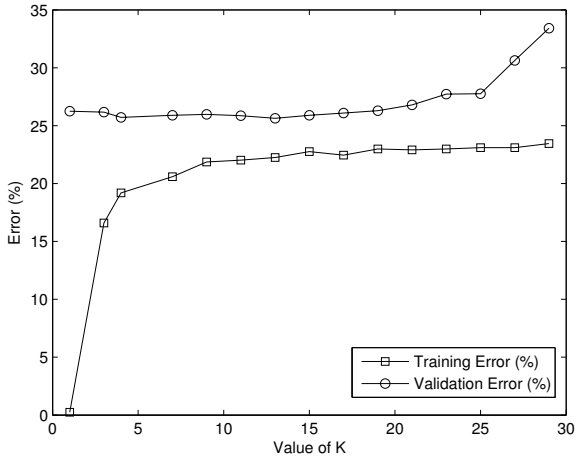


Fig. 2: The learning curve of the K-Nearest Neighbour classifier on the CAST−30 dataset used to determine the optimum value of K.

classifier is important since a too low K value usually generates a *low bias-high variance* classifier which may experience an *overfit*. On the other hand, a too high K value may generate a *high bias-low variance* classifier which perhaps *underfits* the data. To find the correct value of K, we have examined the bias-variance tradeoff using a *learning curve* where training and validation error rates of the K-Nearest Neighbour classifiers are plotted by varying the value of K from 1 to 29; only the odd numbers in this range have been considered. As suspected, the learning curve in Figure 2 illustrates that low K values generate *low bias-high variance* classifiers for the CAST−30 dataset: the classifiers have very low training error but comparatively high validation error. However, according to the curve, we can expect to get a smooth decision boundary with $K = 15$. The K values for the remaining datasets are obtained in a similar way independently for each dataset.

D. Evaluation Measures

To summarize the performances of the classifiers, we have used a wide variety of standard evaluation measures. The measures include precision, recall, F-score, accuracy, false positive rate, false negative rate, area under curve (AUC), and the Matthews correlation co-efficient. Noting that all of the datasets we have used suffer from a high class imbalance ratio (see Table I), we have selected the measures because all except accuracy can deal with the class imbalance problem. We did not use the popular ROUGE [32] evaluation score since we have considered our system performance from the machine-learning classification point of view rather than evaluating the overlapping units of its summary with the reference summary. Another reason to choose classic evaluation metrics like precision, recall, and F-score is for a fair comparison with the contemporary studies that also have considered the same datasets and same evaluation metrics.

IV. Results and Discussions

A. Performance of the Proposed Method

Table III summarizes the performance of our proposed method and shows comparable results from previous studies. The comparisons discussed below are made with a paired *t-test* by setting the significance level, α, to 0.05.

The precision, recall, and F-scores for our method for the two CAST datasets are impressive (Rows 1 and 5). In addition, the high AUC values show that our method does not suffer from any serious *recall-FPR* tradeoff and the *near-perfect* MCC scores indicate that the class imbalance ratio present in these datasets has almost no effect on the learned model. The impressive *FNR* values for the datasets denote that our method misclassifies only 1% and 0.6% of the summary sentences for the CAST−15 and CAST−30 datasets, respectively. As expected, the method performs slightly better on the CAST−30 dataset. For the CAST−15 dataset, our results are compared with the *Head Noun and Verb Model* [15], the *Position and Graph Model* [14], and the *Keyword-based Model* [12] (see Rows 2–4). The data show that our method significantly outperforms them. Similarly for the CAST−30 dataset, we have compared our result with the models proposed by Villatoro-Tello *et al.* [13] (Rows 6 and 7), Leskovec *et al.* [14][15] (Rows 8 and 9), Mitkov *et al.* [12] (Row 10), and Berker and Güngör [11] (Row 11). Again, the data clearly show that our method outperforms all of them by a wide margin. Note that Mitkov *et al.* [12] have reported only the F-score (Rows 4 and 10) while Berker and Güngör [11] have reported only the precision of their methods (Row 11). Overall, the results on the CAST datasets suggest that our method is highly effective for classifying summary sentences from single documents.

On the contrary, the results of our method on the DUC datasets are not as good as we had expected. Table III shows that for these datasets, our method has maintained outstanding recall but has lost ground on precision. As a result, we find comparatively low F-scores on these datasets (Rows 14 and 17). We found negligible differences in the precision, recall, and F-score between the two CAST datasets, but the difference in these scores between the two DUC datasets is more substantial. Although the AUC scores are still high, low MCC scores suggest that our method has suffered from the high class imbalance ratio of the DUC datasets. In addition, the *FNR* values indicate that it misses about 2% and 1% of the summary sentence data from the two datasets. For the DUC−200 dataset, we have compared our results with Leskovec *et al.* [14][15]. With their *Position and Graph* model [14], they reported an F-score of 44.2% (Row 12) while the F-score with their *Heads of Logical Form Triplets* model is 40.0% [15] (Row 13). These results are much better than what we have obtained. Also, the difference between our F-score and that reported by Ferreira *et al.* [16] is not statistically significant (Row 15). However, our method performs significantly better than the Columbia multi-document summarizer [18] for both of the DUC datasets (Rows 16 and 19). Readers are encouraged to examine the results of the top five summarizer systems submitted to the DUC 2002 competition [18]. The summarizer with system code 19 performed the best on the DUC−200 dataset: F-score, 18.4%. On the other hand, the summarizer with system code 21 had the best F-score (25.2%) on the DUC−400 dataset. Therefore, our results are much better than these submitted systems.

Dataset	Row	Method	Precision%	Recall%	F-score%	Accuracy%	FPR	FNR	AUC	MCC
CAST-15	1	**Stylometry Model**	**99.0**	**98.7**	**98.9**	**99.7**	**0.001**	**0.010**	**0.999**	**0.989**
	2	Head Noun and Verb Model	48.0	47.0	48.0					
	3	Position and Graph Model	32.5	70.9	44.5					
	4	Keyword-based Model			32.9					
CAST-30	5	**Stylometry Model**	**99.1**	**99.4**	**99.3**	**99.6**	**0.003**	**0.006**	**0.997**	**0.989**
	6	Word Sequence Model	96.5	84.5	90.1	84.5				
	7	Single Word Model	88.7	84.4	86.5	79.8				
	8	Position, Graph and Linguistic Model	44.5	65.6	53.0					
	9	Heads of Logical Form Triplets Model	42.0	67.0	52.0					
	10	Keyword-based Model			46.3					
	11	Lexical Chain Model	46.0							
DUC-200	12	Position and Graph Model	33.7	64.4	44.2					
	13	Heads of Logical Form Triplets Model	40.0	40.0	40.0					
	14	**Stylometry Model**	**18.0**	**98.4**	**30.4**	**89.4**	**0.108**	**0.017**	**0.983**	**0.397**
	15	Text Graph Model	19.0	62.0	30.0					
	16	Columbia Summarizer	19.0	14.7	16.6					
DUC-400	17	**Stylometry Model**	**42.1**	**98.6**	**59.0**	**93.7**	**0.065**	**0.014**	**0.987**	**0.623**
	18	Text Graph Model	17.0	53.0	25.4					
	19	Columbia Summarizer	23.8	19.6	21.5					

TABLE III: Summary of the performance of the proposed two-stage method using stylometric attributes, as well as results from previous studies.

Also the data show that our F-score is better than the Text Graph Model proposed by Ferreira *et al.* [16] (Row 18) on the DUC-400 dataset. To sum up, the results on the DUC datasets are mixed—the method still lacks the desired precision for multi-document summarization. One possible explanation for these low precision values is the following. When the human summaries are produced from multiple documents, from a set of similar sentences only one (or a few) is usually chosen. Since each cluster in the DUC2002 collection is composed of approximately 10 newswire articles on the same topic, the likelihood of choosing the wrong sentence(s) from a set of similar sentences increases when choosing fewer sentences (DUC-200 vs. DUC-400), thus increasing the number of *false positives*. This hypothesis is supported by the *FPR* values in Table III: the *FPR*s of our method on the DUC datasets (Rows 10 and 14) are significantly higher than those on the CAST datasets (Rows 1 and 4), and the *FPR* is higher for the DUC-200 dataset compared to the DUC-400 dataset.

To find out why the two-stage classification obtained such remarkable results, we need to examine the individual performance of the two learners on the four datasets. Table IV summarizes the *FPR* and *FNR* of the K-Nearest Neighbour and Naïve Bayes learners. These measures have been obtained using a 10-fold cross validation. For the K-Nearest Neighbour learners, the *low FPR-high FNR* in the data refers to their good ability to classify positive data (summary sentences) and their poor ability to classify negative data. This is completely opposite for the Naïve Bayes learners—they have *low FNR-high FPR*. In essence, the two learners are complementing each other thereby resulting in the reported classification results.

Dataset	Learner	FPR	FNR
CAST-15	K-Nearest Neighbour	0.117	0.787
	Naïve Bayes	0.489	0.155
CAST-30	K-Nearest Neighbour	0.239	0.606
	Naïve Bayes	0.503	0.147
DUC-200	K-Nearest Neighbour	0.023	0.969
	Naïve Bayes	0.517	0.161
DUC-400	K-Nearest Neighbour	0.044	0.930
	Naïve Bayes	0.530	0.143

TABLE IV: *FPR* and *FNR* of the K-Nearest Neighbour and Naïve Bayes learners.

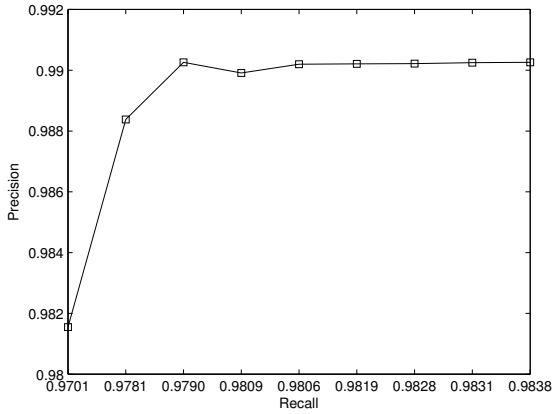
B. Effect of Data Size

In this section, we report the effect of data size on classification performance. To observe this effect, treating each dataset independently, we have generated nine equally sized sets. Each set contains $x\%$ of the original data with repetition and x has been varied from 10% to 90% with an increment of 10% per set. Therefore, our first set contains 10% and the ninth set contains 90% of the data. Then, on each original set we have run a 10 times 10-fold cross validation and recorded our method's precision and recall. With the data found from this experiment, we have analyzed (but not included here) the *precision-recall curve* for each dataset.

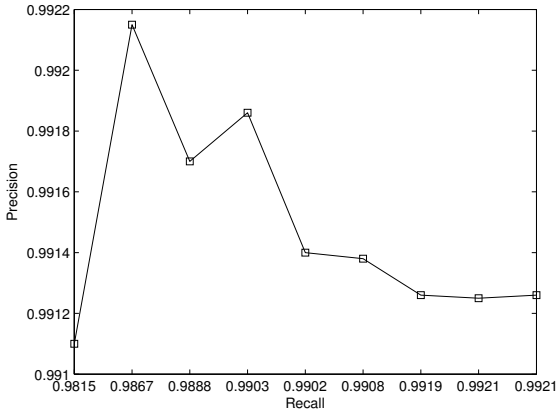
The precision-recall curves in Figure 3 on the the next page show that the data size has some effect on our method's performance. For the CAST-15 dataset, the curve is *ideal* and outlines the increase of both the precision and recall with the growth of the data size (Figure 3a). However, for the CAST-30 dataset, we get better recall with more data by compromising the precision ever so slightly (Figure 3b). On the other hand, the tradeoff is relatively high for the DUC datasets. Increasing data increases the recall slightly with a steep decrease in precision. Most importantly, however, the plots suggest that the precision-recall tradeoff stabilizes as we add more data.

V. Conclusions and Future Work

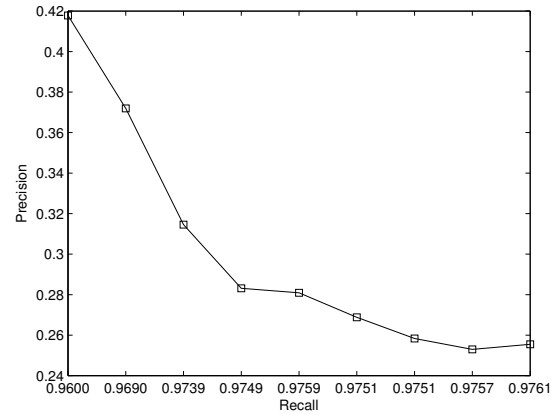
Summary sentence classification is regarded as a pivotal and reasonably complex step to generate summaries for text documents. Importantly, the quality of a summary depends much on the correctness of this step. Our aim is to provide a simple solution to this reasonably complex problem. We propose a statistical learning method to model sentences using a novel set of attributes related to stylometry—a popular linguistic study of writing styles. To learn the models, a discriminative and a generative learner are used in an interesting two-stage classification setup. The learners used in this experiment are



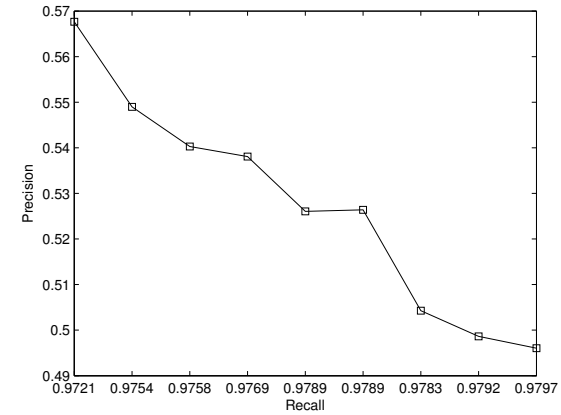
(a) Precision-Recall Curve for the CAST−15 dataset.



(b) Precision-Recall Curve for the CAST−30 dataset.



(c) Precision-Recall Curve for the DUC−200 dataset.



(d) Precision-Recall Curve for the DUC−400 dataset.

Fig. 3: Precision-Recall Curves for the four datasets.

the K-Nearest Neighbour and the Naïve Bayes learners. Our extensive experiments with the summarization data collected from two benchmark datasets named the CAST dataset and the DUC2002 collection strongly suggest that the proposed method performs very well for the single document summarization task. On the other hand, its performance is mixed for classifying summary sentences from multiple documents. Finally, comparisons show that our method performs much better than most other well-known summarization methods.

There is still room for improvement. Our performance on multi-document summarization is mixed. Although we have outlined possible reasons for this in Section IV, a thorough investigation is left as future work. Also, more analysis should be done on anaphora, at least for the DUC datasets, since *pronouns* are regarded as a salient attribute for these datasets. Our results claim only that for summary generation, attributes based on authors' writing style signal text importance much better than the content-based attributes. Our experience has shown that combining various attributes can achieve even more improvement over each individually. Future work can attempt to combine content-based and stylometric attributes.

ACKNOWLEDGMENTS

Support for this work was provided through a Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant to Robert E. Mercer. The authors would like to thank Constantin Orăsan for providing the CAST datasets. The authors are also indebted to the Document Understanding Conference authority for approving the access to the DUC datasets.

REFERENCES

[1] M. Eppler and J. Mengis, "The Concept of Information Overload: A Review of Literature from Organization Science, Accounting, Marketing, MIS, and Related Disciplines," *Kommunikationsmanagement im Wandel*, pp. 271–305, 2008.

[2] D. R. Radev, E. Hovy, and K. McKeown, "Introduction to the special issue on summarization," *Computational Linguistics*, vol. 28, no. 4, pp. 399–408, Dec. 2002.

[3] E. Hovy, "Text summarization," in *The Oxford Handbook of Computational Linguistics*, ser. Oxford Handbooks in Linguistics, R. Mitkov, Ed. Oxford: Oxford University Press, 2003, ch. 32, pp. 583–598.

[4] W. T. Chuang and J. Yang, "Extracting sentence segments for text summarization: A machine learning approach," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '00. USA: ACM, 2000, pp. 152–159.

[5] U. Hahn and I. Mani, "The challenges of automatic summarization," *Computer*, vol. 33, no. 11, pp. 29–36, Nov. 2000.

[6] E. Lloret and M. Palomar, "Challenging issues of automatic summarization: Relevance detection and quality-based evaluation." *Informatica*, vol. 34, no. 1, pp. 29–35, 2010.

[7] T. Givón, *Syntax: A functional typological introduction*. Amsterdam: John Benjamins, 1990.

[8] S. Ji, "A textual perspective on givn's quantity principle," *Journal of Pragmatics*, vol. 39, no. 2, pp. 292 – 304, 2007, focus-on Issue: Discourse, Information, and Pragmatics.

[9] L. Hasler, C. Orăsan, and R. Mitkov, "Building better corpora for summarisation," in *Proceedings of Corpus Linguistics 2003*, UK, March 2003, pp. 309 – 319.

[10] P. Over and W. Liggett, "Introduction to DUC: An intrinsic evaluation of generic news text summarization systems," in *Proceedings of the Document Understanding Conference (DUC2002)*, 2002.

[11] M. Berker and T. Güngör, "Using genetic algorithms with lexical chains for automatic text summarization." in *Proceedings of the 4th International Conference on Agents and Artificial Intelligence (ICAART2012)*. SciTePress, 2012, pp. 595–600.

[12] R. Mitkov, R. Evans, C. Orăsan, I. Dornescu, and M. Rios, "Coreference resolution: To what extent does it help NLP applications?" in *TSD*, ser. Lecture Notes in Computer Science, vol. 7499. Springer, 2012, pp. 16–27.

[13] E. Villatoro-Tello, L. V. Pineda, and M. M. y Gomez, "Using word sequences for text summarization." in *Text, Speech and Dialogue*, ser. Lecture Notes in Computer Science, P. Sojka, I. Kopecek, and K. Pala, Eds., vol. 4188. Springer, 2006, pp. 293–300.

[14] J. Leskovec, N. Milic-Frayling, and M. Grobelnik, "Extracting summary sentences based on the document semantic graph," Microsoft Research, Tech. Rep. MSR-TR-2005-07, January 2005.

[15] J. Leskovec, N. Milic-frayling, and M. Grobelnik, "Impact of linguistic analysis on the semantic graph coverage and learning of document extracts," in *Proceedings of the Twentieth National Conference on Artificial Intelligence(AAAI2005)*, 2005, pp. 1069–1074.

[16] R. Ferreira, L. de Souza Cabral, F. Freitas, R. D. Lins, G. de Frana Silva, S. J. Simske, and L. Favaro, "A multi-document summarization system based on statistics and linguistic treatment," *Expert Systems with Applications*, vol. 41, no. 13, pp. 5780 – 5787, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417414001523

[17] 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.

[18] K. M. David, D. Evans, A. Nenkova, R. Barzilay, V. Hatzivassiloglou, B. Schiffman, and J. Klavans, "The Columbia multi-document summarizer for DUC 2002," in *Proceedings of the ACL Workshop on Automatic Summarization/Document Understanding Conference (DUC2002)*, 2002, pp. 1–8.

[19] K. R. McKeown, V. Hatzivassiloglou, R. Barzilay, B. Schiffman, D. Evans, and S. Teufel, "Columbia multi-document summarization: Approach and evaluation," in *Proceedings of the Document Understanding Conference (DUC2001)*, 2001.

[20] B. Schiffman, A. Nenkova, and K. McKeown, "Experiments in multidocument summarization," in *Proceedings of the Second International Conference on Human Language Technology Research*, ser. HLT '02. USA: Morgan Kaufmann Publishers Inc., 2002, pp. 52–58.

[21] T. Rose, M. Stevenson, and M. Whitehead, "The reuters corpus volume 1 - from yesterdays news to tomorrows language resources," in *Proceedings of the Third International Conference on Language Resources and Evaluation*, 2002, pp. 29–31.

[22] R. Shams and R. E. Mercer, "Classifying spam emails using text and readability features," in *Proceedings of the 2013 IEEE 13th International Conference on Data Mining (ICDM2013)*. USA: IEEE, 2013, pp. 657–666.

[23] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, ser. NAACL '03. USA: Association for Computational Linguistics, 2003, pp. 173–180.

[24] B. Santorini, "Part-Of-Speech tagging guidelines for the Penn Treebank project (3rd revision, 2nd printing)," Department of Linguistics, University of Pennsylvania, USA, Tech. Rep., 1990.

[25] M. B. Kursa and W. R. Rudnicki, "Feature selection with the Boruta package," *Journal of Statistical Software*, vol. 36, no. 11, pp. 1–13, 2010.

[26] R. Nilsson, J. M. Peña, J. Björkegren, and J. Tegnér, "Consistent feature selection for pattern recognition in polynomial time," *Journal of Machine Learning Research*, vol. 8, pp. 589–612, May 2007.

[27] P. A. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features." in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, vol. 1, 2001, pp. 511–518.

[28] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal on Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004.

[29] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, no. 1, pp. 37–66, Jan. 1991.

[30] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, ser. UAI'95. USA: Morgan Kaufmann Publishers Inc., 1995, pp. 338–345.

[31] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[32] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proc. ACL workshop on Text Summarization Branches Out*, 2004, p. 10.