Detecting Human Features in Summaries - Symbol Sequence Statistical Regularity

George Giannakopoulos[1], Vangelis Karkaletsis[1], and George Vouros[2]

[1] Software and Knowledge Engineering Laboratory, National Center of Scientific Research "Demokritos", Greece
ggianna@iit.demokritos.gr, vangelis@iit.demokritos.gr
[2] Department of Digital Systems, University of Pireaus, Greece
georgev@unipi.gr

Abstract. The presented work studies textual summaries, aiming to detect the qualities of human multi-document summaries, in contrast to automatically extracted ones. The measured features are based on a generic statistical regularity measure, named Symbol Sequence Statistical Regularity (*SSSR*). The measure is calculated over both character and word n-grams of various ranks, given a set of human and automatically extracted multi-document summaries from two different corpora. The results of the experiments indicate that the proposed measure provides enough distinctive power to discriminate between the human and non-human summaries. The results hint on the qualities a human summary holds, increasing intuition related to how a good summary should be generated.

1 Introduction

In the domain of natural language processing numerous attempts have been made to identify what is the set of qualities that renders a text understandable and fluent. This problem has been apparent in machine translation (MT), natural language generation (NLG) and automatic summarization. In addition, linguists and tutors have been grading various degrees of fluency of given texts, usually judging L2 authors (authors judged on their second, non-native, language). Within this study we focus on a notion of regularity, apparently a concept related to grammaticality and fluency more than other textual features.

Grammaticality is the quality of conforming to a given grammar. Fluency, on the other hand, is referred to mostly as a measure of text production or reading rate – *i.e.* fluent writers write more quickly that less fluent ones [3] and fluent readers read faster than non-fluent ones. However, the notion of fluency has also been used to describe well-formed, easily understood text [17].

In existing bibliography, there are methods that use grammaticality as a measure of acceptability [1]. Similarly, grammaticality has been considered to be a measure of performance for machine translation systems [9] and summarization systems [6]. The quantification of grammaticallity is a non-trivial task, which has led various researchers towards methodologies concerning the calculation of a grammaticality measure. Most approaches stand upon either parsers [17] or constraint satisfaction models [13].

In summary texts, the grammar considered for grammaticality is that of the used language (*e.g.* English). In our approach, we do not use a given grammar: we use a measure we term Symbol Sequence Statistical Regularity (*SSSR*). This measure indicates, in a graded fashion, whether sequences of symbols (*e.g.* words, characters) have been found to be neighbours in a manner similar to the sequences in a set of given, training texts. With the use of *SSSR*, we try to determine statistically the differences of human and automatic summary texts, to gain intuition required to create better summarization systems.

The rest of the paper is structured as follows. In section 2 we provide background information and a brief review of related works. Then, we present the Symbol Sequence Statistical Regularity definition, along with the proposed methodology for its usage, in section 3. Experiments follow, in section 4, providing the validation of our anaysis. We close the paper with the conclusions and the lessons learned from the study, in section 5.

2 Related Work

In the research quest for fluency and grammaticality evaluation, the works of various linguists and philosophers concerning grammars and acceptability have provided both the foundations and a constant source of research. The work of N. Chomsky for many years has delved upon the notion of grammaticality [4,5]. Both in statistical as well as non-statistical aspects of grammaticality, it has been considered that there can be either a binary decision upon whether a text segment is grammatical, or a graded decision.

Research considering how grammaticality can be graded in a non-binary way has treated grammar as a set of constraints that are either realized or not within a text. There have been distinctions between soft and hard constraints, related to how important a constraint is to the acceptability of a clause [13,22]. Much of the existing work has been based on Optimality Theory (see [21]), which declares that output language is based on a procedure that uses a "candidate analysis" generation function, called *GEN* and a *harmonic* evaluation function, called *H-eval*, which evaluates candidates according to a harmony criterion. The GEN function is part of a Universal Grammar that generates candidate alternatives of analysis for the input, while and the H-eval function is created based on well-formedness rules of a given language. The methodology using GEN and H-eval describes a loop between the two components, until no analysis generated by GEN can give better harmonical results.

In [1], we find an approach based on Property Grammars, which is also a constraint based syntactic formalism. Using Property Grammars tha authors, in the process of analysis, detect the number of properties related to a constituent. Then, these properties are evaluated, and a quantization is performed by measuring the number of violations, non-violations and the number of all the properties evaluated. Highly grammatical text chunks will have a low number of violations for a given set of evaluated properties. The method also applies constraint weighting, which has been used in other works as well [22]. The output is a *grammaticality index* that is shown to correlate to human acceptability evaluations.

From the domain of machine translation, we find the X-Score evaluation process [9], which computes a "target language", consisting of morphology and relationships extracted from a model corpus called "fluency corpus". Then a tagger is used to apply morhological and relation tags to terms of the evaluated text. The assumption used by the authors of X-Score is that the fluency score of a text should be linearly dependent on the frequencies of tags. A prediction function is estimated based on the fluency corpus and then the function is applied to the frequencies of tags of any evaluated text, returning the estimated fluency index of the evaluated text. In the work described in [17] the prediction of acceptability is viewed as a machine learning problem, where the output of a set of parsers is used as input to a learner, trying to discern human from machine generated sentences. Then, the distance of evaluated texts from the support vectors output by the SVM learner determine a metric that correlates to human judgements of fluency.

Here, we should note that a number of other methods of evaluation have been used in the domain of summarization and machine translation, like the ROUGE/BE family of evaluators [14,10] or their "Machine Translation"-related predecessor BLEU [19]. These use word n-grams or sets of terms (*e.g.* head-relation pairs) extracted from the evaluated text and compare them to similarly extracted n-grams or sets of terms from a model corpus. Then, recall and precision related measures can be used to evaluate the given text. These methods however, together with they Pyramid method [20] and the AutoSummENG family of methods [8,7], are mostly meant to evaluate content and not grammaticality.

Studies that analyse human summarization in order to understand the process have been conducted in the past, revealing the abstractive nature of human summarization. In [11], the section on Corpus Analysis indicates that a number of sentences in human summary texts had no equivalent in the source documents. Furthermore, most of the sentences that indeed had an equivalent in the original texts were transformed and combined in various ways to form the summary. Various aspects of the summarization process have been examined in [18] as well, in terms of the processes that define the content and the methodology of rewriting sentences. A series of studies, also apparent in [6], show that automatic systems lack in various domains of text quality, even though they do well in content selection.

The state of the art contains various kinds of evaluators concerning grammaticality and fluency, which are both indicators of acceptability and regularity of text. Our method is related to various of these evaluators, because it: uses a model corpus; derives patterns from the model corpus; uses machine learning methods to discriminate between human and machine-generated texts. The differentiating factors of the presented method, on the other hand, are the following. First, we do not extract a grammar from a text; we determine *regularity*, given an input corpus. This regularity may correlate to various qualities that render a text normal and not only grammar. Second, we do not apply preprocessing of any kind to the input text. Only word splitting is used, *if* we want to use word n-grams. Third, our method requires no underlying language knowledge. This way it functions independently of language. The model of *regularity* is extracted from a given corpus. Fourth, we use sub-word structure information, by the use of character n-grams. Last, the method supports variable granularity, allowing to detect different types of regularity, from word spelling to syntax.

At this point, we want to focus on the fact that the presented study used the analysis methodology as a *means* to derive important lessons. Therefore, we have not studied in full the application spectrum of the analysis method itself.

3 Symbol Sequence Statistical Regularity

In order to analyse the differences between human summary texts and automatically generated ones, we have created a representation of text that includes sequence information. We wanted our representation to be parametric in terms of desired granularity and also allow comparison between its instances. We also wanted to add the ability to cope with fuzziness and ambiguity of expression.

The produced representation, which we call Statistical Symbol Sequence Representation (*SSS-Rep*), is *a set of triples*, including a *pair* and *a corresponding distribution for each pair*. The first part *F* of each pair is a sequence of symbols; each symbol can in fact be either a single letter, a word or a whole sentence. The second part *S* is a single symbol[3]. The distribution *D* for a given pair describes the number of co-occurences of F and S in the text as a function of distance between them, up to a maximum distance d_{max}. This distance is measured as the (integer) number of symbols from *F* to *S* in the text, so if we talk about words the distance is measured in words, if we talk about characters, in characters. From now on we denote such a representation as a set of triplets in the form: $F \rightarrow S(D)$, where $D \equiv$ (distance1 $\Rightarrow$ numberOfOccurences1distance2 $\Rightarrow$ numberOfOccurences2...) in a sparse distribution representation. $D(x)$ identifies the number of occurences for a given distance x. We cosider $\mathbb{D}$ to be the powerset of sparse distribution representations. So, if distance1 is 1 and distance2 is 4 and their numberOfOccurences are 2, 5 correspondingly, then we have found 2 times *S* to be in a distance of 1 from *F* in the text and 5 times in a distance of 4.

The *SSS-Rep* has a set of parameters, indicative of its granularity and fuzziness: the r-gram rank *r* of *F*, the maximum distance d_{max} of co-occurence as well as the type of symbol (*e.g.* character, word, sentence). Thus, we will use the form *SSS-Rep*(r, d_{max}, *symbolType*) to fully describe an *SSS-Rep* and we will call every triplet that can be derived from a given *SSS-Rep an instance of* SSS-Rep.

Example 1. The sentence:
"A_big_big_test."
is represented as *SSS-Rep*(1,1,character) by:
$t \rightarrow e(1 \Rightarrow 1.0); \_ \rightarrow b(1 \Rightarrow 2.0); A \rightarrow \_(1 \Rightarrow 1.0); \_ \rightarrow t(1 \Rightarrow 1.0); b \rightarrow i(1 \Rightarrow 2.0); t \rightarrow .(1 \Rightarrow 1.0); e \rightarrow s(1 \Rightarrow 1.0); g \rightarrow \_(1 \Rightarrow 2.0); s \rightarrow t(1 \Rightarrow 1.0); i \rightarrow g(1 \Rightarrow 2.0)$
while in *SSS-Rep*(2,2,word) by:
$big, big \rightarrow test(1 \Rightarrow 1.0); a, big \rightarrow test(2 \Rightarrow 1.0); a, big \rightarrow big(1 \Rightarrow 1.0)$

[3] Using a single symbol in the second part allows efficient calculation of the *SSS-Rep*.

We say that the first set of triplets is an *instance of SSS-Rep*(1,1,character), while the second an *instance of SSS-Rep*(2,2,word).

We note that, essentialy, *SSS-Rep*(1,1,word) can directly map to a bigram language model [15]. Thus, some *SSS-Rep* configurations are n-gram language models. However, in general *SSS-Rep* configurations are not n-gram models. The study of the exact relation between *SSS-Rep* and n-gram models is, however, outside the scope of this work.

Having defined *SSS-Rep*, we define a measure of similarity between two instances of an *SSS-Rep*, given the fact that they share the *same parameters*. First we prepare our functional tools:
If T_1, T_2 are two instances of *SSS-Rep*(r, d_{max}, *symbolType*), we define the membership operator:

$$
\begin{aligned}
&\textit{SSS-Rep}(r, d_{max}, \text{symbolType}) \vdash T_1 \iff \\
&T_1 \dashv \textit{SSS-Rep}(r, d_{max}, \text{symbolType}) \iff \\
&T_1 \text{is an instance of } \textit{SSS-Rep}(r, d_{max}, \text{symbolType}) \iff \\
&T_1 = \{(x, y, z) : \text{isA}(y, \text{symbolType}), \\
&\text{isNGramOf}(x, \text{symbolType}), \text{rank}(x) = r, \\
&\{z\} \in \mathbb{D}\}
\end{aligned}
$$

where rank(x) gives the n-gram rank of x, and isA(x, symbolType) returns true if x is a symbol of type symbolType and isNGramOf(y, symbolType) returns true if y is a sequence of symbols $y = \{y_1, y_2, ..., y_n\}, n \in \mathbb{N}^* : \forall 1 \leq i \leq n, i \in \mathbb{N}, \text{isA}(y_i, symbolType)$.

We also define a membership function connecting triplets to an *SSS-Rep* instance T:
$A \equiv (F \rightarrow S(D)) \in T \iff$ there exists an identical triplet A in the set of triples representing T.

We define a similarity measure sim between distributions D_1, D_2, even though existing similarity measures like KL-divergence or chi-square can be used as well. We used a simple sum of absolute differences, because it cost less processing time to calculate. In fact the similarity between two distributions is the sum of the absolute differences *for all the non-zero elements for either distribution*. If X are the values of $\{x : D_1(x) > 0 \text{ or } D_2(x) > 0\}$, then: $sim(D_1, D_2) = \sum_{i \in X}(abs(D_1(i) - D_2(i)))$, where abs is the absolute value function.

On the same basis, the similarity of two triplets A, A', simT(A, A') equals to the similarity of their distributions D, D', sim(D, D') if the two first elements of the triples are identical, else we define simT(A, A') $= 0$. For T_1, T_2,
$T_1 \dashv$ *SSS-Rep*(r, d_{max}, symbolType),
$T_2 \dashv$ *SSS-Rep*(r, d_{max}, symbolType) we define the regularity function of T_1 given T_2 and its corresponding operator "$\sim$ ":

Definition 1. $T_1 \sim T_2 \equiv$
$regularity(T1|T2) = \frac{\sum_{A \in T_1, A' \in T_2} sim(A, A')}{|T_1|}$, *where* $|T_1|$ *is the number of triplets in* T_1. *We use the* $|T_1|$ *in the denominator because* T_1 *is the triplet judged and* T_2 *the a-priori evidence: we need the regularity function to be anti-symmetrical.*

In order to define the *SSS-Rep* of a corpus $C = T_1, T_2, ..., T_n, n \in \mathbb{N}^*$, we can simply concatenate the corpus texts in a single super-text. Given this definition, we can also define the comparison between a corpus and a text, by comparing the corresponding super-document and the document *SSS-Rep*s. This is what we call *SSSR*.

4 Experiments

The data on which our experiments were conducted were the summary and evaluation corpus of DUC 2006. The corpus consists of summaries for 50 different topics, as well as the corresponding 25 input

documents per topic from which the summaries were generated. Each topic had a number of automatically extracted summaries, one for each participating system, and 4 human created summaries. The human summaries were differentiated by means of an identifier, as were the baseline system summaries, which originated from a baseline system created by NIST, which simply took the first 250 words of the most recent document for each topic. All summaries were truncated to 250 words before being evaluated. To verify some of our experiments using a second corpus, we have used the corpus of DUC 2007 as well. The corpus, similarly to the one of DUC 2006, consists of summaries for 45 different topics. All topics had 4 human summaries each, as well as 28 machine generated summaries. In the corpora the human summaries appeared both as models and peers (i.e., twice each). In this study we have kept both duplicates of human summaries as "human" instances, so the count of human summaries will appear to be double the expected.

In order to use baseline-quality texts, we created a single automatic summary for each topic in the DUC2006 corpus. The summary was created by randomly adding words from the 25 input documents in a way that the statistics of the words (frequency) would tend to be the same as the input documents. The length of the summaries is about 250 words (length chosen from a Poisson distribution averaging to 250).

To determine whether the presented methodology extracts features that discern between human and automatic summaries, we have conducted the following process over two different corpora. For a given topic, the set of input documents were analysed to determine the $\{$*SSS-Rep*$(i, j, character), 1 \leq i \leq 8, j = i\}$ set of representations for each topic. We also performed word analysis to get the set of representations $\{$*SSS-Rep*$(k, l, word)$, where $1 \leq k \leq 3, l = k\}$. Both character and word *SSS-Rep* created the representation $C_{\textit{SSS-Rep}}$ of the analyzed topic. For each summary document, either human or automatically-generated, we extracted the same set of representations as those of the corresponding topic. We compared each summary text representation $T_{\textit{SSS-Rep}}$ to the corresponding topic representation $C_{\textit{SSS-Rep}}$, creating a feature vector, the values of which were the results of $T_{\textit{SSS-Rep}} \sim C_{\textit{SSS-Rep}}$ for the *SSS-Rep* configurations. This gave a 11-dimensional vector, which we matched to a label $L \in \{human, peer\}$. The "peer" label was assigned to automatic summarizer documents plus our baseline documents. We used a simple Naive Bayes and a kernel-estimating Naive Bayes classifier [12] to determine whether the vectors were enough to classify human and non-human (peer) texts effectively. In both cases 10-fold stratified cross-validation was performed to determine the effectiveness of the method (see the WEKA toolkit [23]). Then, we used an SVM classifier as well, to validate the results. We calculated each feature's Information Gain and performed Principal Component Analysis to determine important features.

4.1 Classification Between Human and Machine-Generated Summaries

The naive Bayes classification using the full feature vector managed to provide an F-measure that exeeds 90%, for both classes. It is impressive that we need not apply more complex classifiers (like SVM or neural networks); this identifies the features used as appropriate. In Table 1 we see the breakdown of correctly classified and misclassified instances. We see that the count of automatic summary texts is much higher than the one of human summaries. This would be expected to lower the effectiveness of the simple naive Bayes; but that has not happened. Once more, the features are shown to be appropriate. We apply a more complex classifier to see if it is easy to maximize the attained F-measure. Using Multinomial Bayes [16], as well as an Radial Basis Function kernel SVM classifier (C-SVM) with a high cost parameter (LibSVM implementation [2]) we got the results shown in Table 1.

At this point we wanted to check whether the SVM model we produced was only overfitting the DUC 2006 corpus data. Thus, we evaluated the model on the DUC 2007 corpus data. Impressively, the results were quite similar, as can be seen in Table 2, amounting to an F-measure of over 97% for both classes. Then, we evaluated the model of DUC 2007 on the DUC 2006 data. The results of this experiment, which are described in Table 2, show that we had an increased number of false negatives for the Human class. This is probably the effect of not including baseline texts in the experiments conducted on the corpus

Table 1. Confusion Matrix — DUC 2006: (left to right) Naive Bayes (NB), Multinomial NB, C-SVM

Classified As		
Peer	Human	Actual Class
1740	60	Peer
13	387	Human

Classified As		
Peer	Human	Actual Class
1780	20	Peer
4	396	Human

Classified As		
Peer	Human	Actual Class
1797	3	Peer
1	399	Human

of DUC 2007, which reduced available information on negatives. In any case, the application of the extraction and learning process by itself yields comparable results for both corpora.

Table 2. Confusion Matrix — C-SVM model: DUC2006 on DUC2007 (left); DUC2007 on DUC2006 (right)

Classified As		
Peer	Human	Actual Class
1439	1	Peer
18	342	Human

Classified As		
Peer	Human	Actual Class
1797	3	Peer
335	65	Human

The experimental results, therefore, illustrated that the use of *SSS-Rep* as the means to represent the corpus and the texts, along with the use of *SSSR* for the extraction of regularity, provide good enough features to tell human and automatically generated summaries apart.

4.2 Feature Importance

At this point, we wanted to see, given the success of the classification process, what the key features in the classification are. We used two methods to decide upon the answer. First, we ranked the features according to their Information Gain (see [15, p. 583] for linguistic uses of the measure), concerning the human-peer classification. Second, we performed Principal Component Analysis [24] to extract complex features that hold the most useful pieces of information.

The information gain calculation gave the ranking of Table 3. In the table, attributes are named according to the *SSS-Rep* used, where the first part ("char" or "word") indicates what kind of symbol was used and the second part what was $r = d_{max}$ paramteter value was. For example, Char2 indicates character symbols with $r = d_{max} = 2$. The Table presents both ranking for DUC 2006 and 2007 corpus on the left and right part correspondingly.

Table 3. Feature Importance Based on Information Gain (left), PCA analysis (right)

Rank	*IG 2006*	SSS-Rep	*IG 2007*	SSS-Rep
1	0.6528	Char8	0.6769	Char7
2	0.6527	Char7	0.67525	Char8
3	0.6463	Char6	0.67394	Char6
4	0.6161	Char5	0.61962	Char5
5	0.3703	Char4	0.35862	Char4
6	0.0545	Char3	0.06614	Char3
7	0.0256	Word3	0.01098	Char1
8	0.0196	Char1	0.0078	Char2
9	0.0133	Word1	0	Word2
10	0	Word2	0	Word3
11	0	Char2	0	Word1

Corpus	*Eigenvalue*	*Feature Formula*
DUC 2006	5.62218	0.414Char4+0.409Char5 +0.389Char6
DUC 2007	5.70926	-0.411Char4-0.397Char5 -0.372Char6

The application of PCA on both the corpora of DUC 2006 and DUC 2007 brought a pleasant surprise: the most important Principal Components (according to their eigenvalue) extracted from both corpora were very similar. Both the absolute values of weights of the original features in the complex features, as well as the eigenvalues of the major principal components themselves were similar (see Table 3), giving high importance to non-word units (character n-grams). This indicates emergent important features, only partially dependent on the corpus.

It seems that the low-ranked character n-grams simply reproduce the spelling constraints of a language and offer no useful information. The most important features appeared to be high-ranked character n-grams, because those overlap more than one word. These features are the ones detecting word collocations and other similar phenomena. Using *only Char7 and Char8 features* with Multinomial Naive Bayes we reached a performance of 99% accuracy (only 16 misclassified peer texts and 8 misclassified human, on a whole of 2176 texts). In Figure 1, the light colored (yellow) areas indicate human instances and the dark colored (blue) peer instances. We see that higher rank character n-grams discriminate between classes: humans have *lower SSSR* in high ranks than automatic summaries, but *higher SSSR* than random texts.

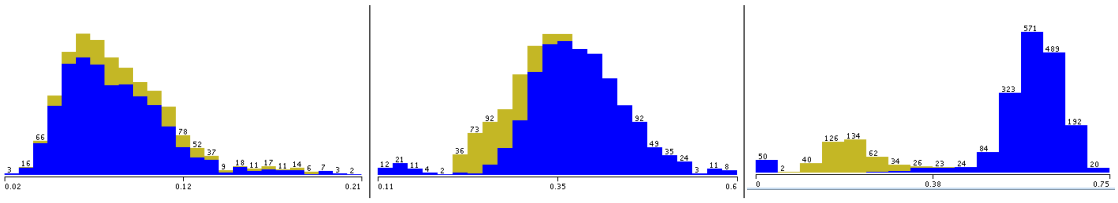

Fig. 1. Distribution of (left to right) character uni-grams, 4-grams, 8-grams *SSSR* for DUC 2006

What is easily noted from the above is that importance is focused in sub-word (character) features. However, it is not the spelling that makes the difference, but the joining of words. We interprete the results as an indication that regularity is not a measure of quality: it is mostly a measure of whether a text is result of an abstraction or reformulation process. That is why people have lower *SSSR* performance than automatic summarization systems, but higher that random texts.

5 Conclusions and Future Work

From the study presented we have inferred a number of facts, concerning mostly the summarization process. First, many existing automatic summarization systems, which are based mostly on extractive techniques, appear to share statistical features. There is such a feature that can tell human summaries apart from automatically generated ones. This is called Symbol Sequence Statistical Regularity, *SSSR*. Second, human summaries tend to have *lower SSSR* values than automatically generated summaries. This may be directly connected to the abstractive nature of multi-document summarization. On the other hand, human summaries tend to have *higher SSSR* values than summaries generated as random text. It would be better, however, if more research was conducted as to what values of *SSSR* would text generated by language models like HMM have. Would *SSSR* remain useful then? Last, the principal components, based on *SSSR*, that discriminate human from automatically-generated summaries for a given language seem to be rather specific. This indicates that humans do follow statistically tracable patterns of text generation if we get to the character level.

In an effort to evaluate automatic texts, with respect to human perception of fluency and grammaticality, the presented *SSSR* measure adds one more scientific tool, which holds such abilities like language-neutrality and objectivity. It would be very important to determine other, perhaps similar measures that

will be able to detect other aspects of human texts. This includes existing n-gram-based methods which can be leveraged to increase our intuition of the complex process of summarization. Such intuition will in turn, hopefully, give birth to better automatic summarization systems. In our research, we have begun to utilize *SSSR* in the process of sentence reformulation for summarization and expect results soon.

The corpora of DUC 2006, 2007 were kindly provided by NIST and have been used according to the ACQUAINT directions. The whole range of experiments for this paper, as well as the editing process and statistical analysis were conducted on Open Source software. The JINSECT toolbox used in the paper can be found at http://sf.net/projects/jinsect.

References

[1] Blache, P., Hemforth, B., Rauzy, S.: Acceptability prediction by means of grammaticality quantification. Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL pp. 57–64 (2006)
[2] Chang, C., Lin, C.: LIBSVM: a library for support vector machines. Software available at http://www. csie. ntu. edu. tw/cjlin/libsvm 80, 604–611 (2001)
[3] Chenowith, N., Hayes, J.: Fluency in Writing: Generating Text in L1 and L2. Written Communication 18(1), 80 (2001)
[4] Chomsky, N.: Grammaticality in the Logical Structure of Linguistic Theory (1955)
[5] Chomsky, N.: Rules And Representations. Columbia University Press (2005)
[6] Dang, H.: Overview of DUC 2006. In: Proceedings of HLT-NAACL 2006 (2006)
[7] Giannakopoulos, G., Karkaletsis, V.: Summarization system evaluation variations based on n-gram graphs. In: TAC 2010 (2010)
[8] Giannakopoulos, G., Karkaletsis, V., Vouros, G., Stamatopoulos, P.: Summarization system evaluation revisited: N-gram graphs. ACM Trans. Speech Lang. Process. 5(3), 1–39 (2008)
[9] Hamon, O., Rajman, M.: X-Score: Automatic Evaluation of Machine Translation Grammaticality. Proceedings of the 5 thInternational Conference on Language Resources and Evaluation (LREC) (2006)
[10] Hovy, E., Lin, C., Zhou, L., Fukumoto, J.: Basic Elements (2005)
[11] Jing, H.: Using hidden Markov modeling to decompose human-written summaries. Computational Linguistics 28(4), 527–543 (2002)
[12] John, G., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: Proceedings of the eleventh conference on uncertainty in artificial intelligence. vol. 1, pp. 338–345. San Mateo (1995)
[13] Keller, F.: Gradience in Grammar. Ph.D. thesis, University of Edinburgh (2000)
[14] Lin, C.: Rouge: A Package for Automatic Evaluation of Summaries. Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004) pp. 25–26 (2004)
[15] Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. The MIT Press (1999)
[16] McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. In: AAAI-98 workshop on learning for text categorization. vol. 752, pp. 41–48 (1998)
[17] Mutton, A., Dras, M., Wan, S., Dale, R.: GLEU: Automatic Evaluation of Sentence-Level Fluency. Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics pp. 344–351 (2007)
[18] Nenkova, A.: Understanding the process of multi-document summarization: content selection, rewriting and evaluation. PhD in Philosophy, Columbia University (2006)
[19] Papineni, K., Roukos, S., Ward, T., Zhu, W.: BLEU: a method for automatic evaluation of machine translation. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics pp. 311–318 (2001)
[20] Passonneau, R., McKeown, K., Sigelman, S., Goodkind, A.: Applying the Pyramid Method in the 2006 Document Understanding Conference (2006)
[21] Prince, C., Smolensky, P.: Optimality Theory: Constraint Interaction in Generative Grammar. Optimality Theory in Phonology: A Reader (2004)
[22] Sorace, A., Keller, F.: Gradience in linguistic data. Lingua 115(11), 1497–1524 (2005)
[23] Witten, I., Frank, E., Trigg, L., Hall, M., Holmes, G., Cunningham, S.: Weka: Practical Machine Learning Tools and Techniques with Java Implementations. ICONIP/ANZIIS/ANNES pp. 192–196 (1999)
[24] Wold, S.: Principal component analysis. Chemometrics and Intelligent Laboratory Systems 2(1), 37–52 (1987)