



HAL
open science

A coherence model for sentence ordering

Houda Oufaida, Philippe Blache, Omar Nouali

► **To cite this version:**

Houda Oufaida, Philippe Blache, Omar Nouali. A coherence model for sentence ordering. NLDB-2019, 2019, Manchester, United Kingdom. 10.1007/978-3-030-23281-8 . hal-02299211

HAL Id: hal-02299211

<https://hal.science/hal-02299211>

Submitted on 27 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A coherence model for sentence ordering

Houda Oufaida¹, Philippe Blache², and Omar Nouali³

¹ Ecole Nationale Supérieure d'Informatique ESI, Oued Smar, Algiers, Algeria,
h_oufaida@esi.dz

² Aix Marseille Université, CNRS, LPL UMR 7309, 13604, Aix en Provence, France,
blache@lpl-aix.fr

³ Centre de Recherche sur l'Information Scientifique et Technique CERIST, Ben
Aknoun, Algiers, Algeria,
onouali@cerist.dz

Abstract. Text generation applications such as machine translation and automatic summarization require an additional post-processing step to enhance readability and coherence of output texts. In this work, we identify a set of coherence features from different levels of discourse analysis. Features have either positive or negative input to the output coherence. We propose a new model that combines these features to produce more coherent summaries for our target application: extractive summarization. The model use a genetic algorithm to search for a better ordering of the extracted sentences to form output summaries. Experimentations on two datasets using an automatic coherence assessment measure show promising results.

Keywords: coherence features · coherence model · sentence ordering · automatic summarization · genetic algorithm.

1 Introduction

Coherence and cohesion are key elements for text comprehension [1]. Coherence involves logical flow of ideas around an overall intent. It reports a conceptual organization of discourse and can be observed at the semantic level. Coherence is essential to text comprehension. Indeed, with a lack of coherence, the text loses quickly its informational value.

Dealing with text coherence remains a difficult issue for several NLP applications such as machine translation, text generation and automatic summarization. Most of automatic summarization systems rely on extractive methods which extract complete sentences from source texts to form summaries. This ensures that the summary is grammaticality correct but in no case its coherence. Considering coherence of extractive summaries involves dealing with sentence informativness input against summary's flow. Several elements contribute to text coherence such as discourse relations [2], sentences connection by mean of common entities patterns [3] and thematic pregression [4].

In the automatic summarization task, it is fundamental to generate intelligible summaries. Extractive techniques succeed in selecting most relevant information

but mostly fail to guarantee their coherence. Only few of these techniques considered coherence as an additional feature in the summary extraction process. It is a difficult task which tackles with multi level discourse analysis: syntactic level which connectors are used to improve text cohesion, semantic level in which textual segments are regrouped around common concepts and finally, global level in which sentences are presented in a logical flow of ideas.

In this paper we deal with coherence as an optimization problem. We identify a set of coherence features that have positive or negative impact on summaries coherence. The intuition is that positive input features such as original thematic ordering in the source text/texts and shared entities of adjacent sentences contribute to local and global coherence. These features should be maximized and negative input features such as redundancy should be minimized.

The rest of the paper is organized as follows: we first introduce a review of the very few works in the field. Second, we describe how our coherence model combines between coherence features to better ordering sentences within system summaries. Details and discussion of our experiments are presented, the coherence model is introduced as a post processing step. Finally, we conclude our work with some interesting perspectives.

2 Related work

Early approaches of automatic summarization use sentence compression techniques to improve summaries' coherence. The main idea is to reproduce human summarization process, namely: i-identify relevant sentences ii-compress and reformulate relevant iii-reorder sentences iv-add discourse elements to make a cohesive summary.

Probably the most referenced work is Rhetorical Structure Theory (RST) discourse analysis [2]. A set of discourse relation markers from an annotated corpus are used to define two elements for each relation: nucleus and satellite. The analysis generate a tree in which the nucleus parts of the top levels are the most relevant ones. [5] train an algorithm on collections of (texts, summaries) to discover compression rules using a noisy-channel framework. The assumption is that the compressed form is the source of a signal which was affected by some noise, optional text. The model learns how to restore the compressed form and assesses the probability that it is grammaticality correct . More recently, [6] define the concept of textual energy of elementary discourse units. It reflects the degree of each segments informativeness: the more the segment shares words with other segment the more it is informative. Less informative segments are eliminated and the remaining segments grammaticality is estimated by mean of a language model.

[4] study the thematic progression in the source texts and identify which thematic ordering is better for the output summaries. The authors define three strategies for sentence ordering: (1) majority ordering which is a generalization of ordering by sentence position and reflects, for each couple of themes, how many source texts sentences from the first theme precede the sentences from

the second one (2) chronological ordering in which themes are ordered by their publication date and (3) Augmented ordering which add a cohesion element that regroups themes whose sentences appear in the same blocks of texts. Sentences in the output summary are assigned to themes and follow the thematic ordering. Augmented ordering seems to be the best alternative for news articles.

[3] define local coherence as a set of sentence transitions required for textual coherence. An entity-based representation of the source text is used to model coherent transitions. The intuition is that consecutive segments (sentences) about same entities are more coherent. The model estimates transition patterns probabilities from a collection of coherent texts.

More recently, [7] introduce a joint model that combines between coherence and sentence salience in the sentence extraction process. A discourse graph is first generated in which vertices correspond to sentences and positive edges weights to coherent transitions between each couple of sentences i-e the second sentence could be placed after the first sentence in a coherent text. It is based on syntactic information such as deverbal noun reference, event/entity continuation and RST discourse markers.

The success of deep learning architectures in various NLP tasks including coherence models was recently investigated. [8] train a three level neural network to model sentences composition to form coherent paragraphs. Here, positive examples are coherent sentence windows and negative examples are sentences windows in which a sentence was randomly replaced. Sentence vectors are induced from the sequence of its word embeddings using recurrent neural networks. The neural network is trained using pairs of original articles and randomly permuted sentences, window size is three consecutive sentences. [9] propose to generalize the entity based coherence model initially proposed by [4] using a neuronal architecture. The model maps grammatical roles within entity grid to a continuous representation (a real valued vector learned by back propagation). Entity transition representations of a given sentence sequence are used by convolution, pooling and linear projection layers to finally compute a coherence score. The model is trained on a set of ordered coherent/less coherent document pairs and compared to several coherence models for three tasks: sentence ordering and summary coherence rating.

In the previous work, various features are used to improve output coherence. RST discourse analysis is certainly of value to define a global coherence model. However, it requires deep text analysis which is not available for most languages. In this work, we have selected a set of coherence features. Each feature is supposed to help the model to give higher or lower coherence score according to a particular sentence ordering. The model combines between features and selects an ordering that maximises the coherence score. We assume that these features, once applied together, complement each other and lead to better coherence. We use genetic algorithm to select a coherent ordering. The advantage is that the model can be easily alimented by additional and language specific features. Features can be added to the fitness function by specifying its contribution to the

output ordering. The next section describes, in detail, the proposed coherence model.

3 Coherence model

In our coherence model, we propose to combine state-of-the-art features using a genetic algorithm. These features are domain independent and could be automatically extracted for a large number of languages.

3.1 Coherence features

Positive input features positive input features are features who should be maximized in the output summary. They are assumed to help the model to produce more coherent summaries.

Sentence position: sentence position feature is based on the assumption that sentence ordering in source text is coherent and a coherent summary should follow the initial ordering. In multi-document summarization, this ordering is generalized using publication date in a way that the first sentence in the first document is given the label "1" and the last sentence in the most recent document is given the label "n", "n" being the number of sentences in all source documents.

Shared entities: it is an important feature based on the assumption that sentences discussing same entities should appear in the same textual segment. [10] defines textual continuity as "a linear progression of elements with strict recurrence" which puts forward that coherent development of text should not introduce a sudden break.

Shared entities feature was introduced by [3], it requires part of speech tagging. In practice, noun phrases tag set depends on target language and the Part of Speech tagger used (NN, NNP, NNS, NNPS, etc. for English Penn Treebank tag set).

We use the number of shared noun phrases between each couple of adjacent sentences in the candidate summary as a positive input feature (1) (2).

$$Common_Entities(S_1, S_2) = \frac{2 \times |Entities(S_1) \cap Entities(S_2)|}{|S_1| + |S_2|} \quad (1)$$

$$Score_Entities(R) = \sum_{i=1..|R|-1} Common_Entities(S_i, S_{i+1}) \quad (2)$$

Thematic ordering: thematic progression is a key factor in information ordering and text comprehension. Presenting information in a logical progression is important especially in summaries which are size limited. Following [4], we want to make summaries thematic progression similar to source texts. We define a precedence matrix (PM) of topics. Each entry $PM[c_i, c_j]$ corresponds to the

percentage of sentences from topic i which appears before sentences from the second topic j in source texts.

Topics	T_0	T_1	T_2	T_3	T_4	T_5
T_0	0.000	0.335	0.285	0.564	0.631	0.521
T_1	0.665	0.000	0.438	0.764	0.787	0.782
T_2	0.715	0.562	0.000	0.865	0.858	0.867
T_3	0.436	0.236	0.135	0.000	0.594	0.486
T_4	0.369	0.213	0.142	0.406	0.000	0.437
T_5	0.479	0.218	0.133	0.514	0.563	0.000

Different possible strategies for thematic ordering could be considered. A first strategy is to order topics according to their precedence value. We define precedence value of a target topic as the sum of remaining topics precedence value to the target topic (sum per column) (3). Topic with minimum precedence will be the first topic to be mentioned in the summary thematic ordering.

$$Precedence_Score(C_j) = \sum_{i=1..|C|} Precedence(C_i, C_j) \quad (3)$$

Another strategy is to build thematic ordering gradually. The algorithm starts with couple of topics with a strong precedence score (T_2 and T_5 in the example). Then the algorithm search for another couple of topics that maximizes precedence scores for the just selected topics at the beginning/end of the previous ordering. Algorithm 1 repeats these steps until finding a complete ordering which includes all topics. We compare system summary ordering against source

1: **Input:**

$Precedence[,]$: precedence matrix

2: **Initialise:**

$Ordering = \{\}$

3: $Ordering = (C_{Max_i}, C_{Max_j}) = Max\{Precedence(C_i, C_j), \forall i, j < |C|\}$

4: **do**

5: $Max_i = Max\{Precedence(*, C_j), \forall j < |C|\}$

6: $Max_j = Max\{Precedence(C_i, *), \forall i < |C|\}$

7: $Ordering = Ordering \cup \{(C_{Max_i}, C_j)\}$

8: $Ordering = Ordering \cup \{(C_i, C_{Max_j})\}$

9: **while** $|Ordering| < |C|$

10: **Return:** Ordering R

Algorithm 1: Pseudo algorithm for thematic ordering extraction

texts thematic ordering using using the distance between the two ordering vectors (4). System summary is likely to be not complete, we complete the shortest

vector by the value of the last item (last topic number)

$$Thematic_Ordering_Score = \frac{1}{Distance(Sum_Ord, Source_Ord)} \quad (4)$$

Negative input features

Redundacy: in addition to the size constraint, redundancy is not recommended. Bringing new information in each sentence is essential to the semantic coherence of any text. In the context of automatic summarization, it is critical to present new relevant information in each single sentence. We use a sentence similarity measure proposed in [11] to compute sentence relatedness between each couple of sentences.

$$Sim(S_1.S_2) = \frac{\sum_i Match(w_i) + \sum_j Match(w_j)}{|S_1| + |S_2|} \quad (5)$$

We define a redundancy score for each system summary as the sum of all relatedness scores of included sentences (6). This feature is competing with the continuity defined by the shared entities feature. Indeed, if two sentences mention the same entities, they are similar to a certain degree.

$$Redundancy_Score = \sum_{i,j=1..|R||i \neq j} Relatedness(S_i, S_j) \quad (6)$$

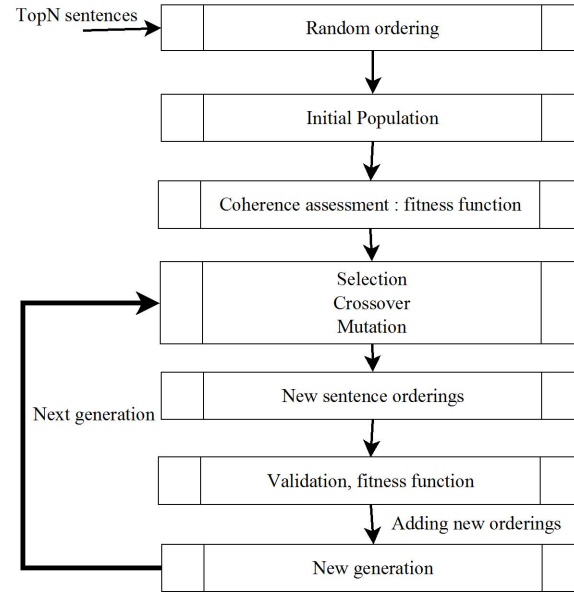
3.2 Coherence model

Our problem is to order most relevant sentence in most possible coherent way. We have defined a set of positive/negative input features that improve/degrade summary coherence. Obviously, evaluating a coherence score for each possible ordering is not feasible. Indeed, a summary of 250 words in English contains approximately 13 to 17 phrases (A sentence contains, in average, 15 to 20 words). In the fitness function, each coherence feature is an objective to be attended (maximize or minimize) in the output summary ordering. Figure 1 presents an overview of the coherence model steps.

Model parameters

Fitness Function: each coherence feature is integrated to the fitness function according to its sense of contribution. For example, (Shared entities, +), (Thematic ordeing, +), (Sentence similarity, -1) is a fitness function. We define several possible combinations and evaluate coherence for each target fitness function.

Ordering codification: each candidate summary ordering is represented by a vector of sentences IDs. Vector size is equal to the number of sentences included in the system summary with respect to the summary's size.



Genetic algorithm steps

Fig. 1: Coherence model

Initial Population: the process of searching the best coherent ordering begins with a random ordering of selected sentences . Each solution is evaluated using the fitness function.

Coherence assessment: Each feature value is calculated for each ordering (chromosome) in the population. An ordering is better than another if it has higher feature values.

Selection: it consists of selecting best coherent ordering from the population to form the next generation. Each ordering which fits the best fitness function (coherence features) is more likely to be selected in the next generation. We use the tournament selection method since it tends to converge quickly towards satisfactory output [12]. Each selected ordering will be a parent of the next generation orderings. Tournament selection is repeated n times until having the complete set of parents.

Crossover: the parents are used to form new orderings using the crossover operator. Two parents are randomly selected and a two-point crossover operator is applied to merge parts of parents and form new orderings. We believe that two points crossover is sufficient for summaries (less than 20 sentences for a summary of 250 words).

Crossover operation may generate invalid orderings in the case of duplicate sentences or surpassed size of desired summary. In this case, invalid children are ignored and the crossover operation is repeated until the desired number of orderings is reached.

Mutation: it consists of randomly switching couple of sentences in the target ordering to create a new one. Besides the crossover operator, mutation assists in genetic diversity. It does not generate invalid summaries since it keeps the same sentences.

Final output: the purpose of the development stage is to make sentence orderings more coherent across generations until reaching the maximum number of generations to be explored. Here, the ordering which fits, the most, fitness function is selected from the last generation as the final output.

4 Experimentation

The main goal of the experimentation is to assess the input of each coherence feature to enhance output coherence. We have implemented our solution under DEAP Package [13] which implements a set of evolutionary algorithms for optimisation problems: genetic algorithms, particle swarm optimization and differential evolution. We have opted for a dynamic fitness function that allows users to define couples of (feature, input sense) to be considered.

4.1 Coherence assessment

It is a difficult task to assess text coherence from different levels; local and global coherence and in all its aspects: rhetorical organization, cohesion and readability. Using a coherence metric is a first quick option to assess coherence features input.

We use Dicomer metric [14] which is based on a model that captures statistical distribution of intra and inter-discourse relations. The model uses a matrix of discourse role transitions of terms from adjacent sentences. The nature of transition patterns and their probability are used to train an SVM classifier. The classifier learns how to rank original texts and texts in which sentence ordering is shuffled. Three collections of texts and summaries from TAC conferences are used to train the classifier.

4.2 Datasets

Since our target task is text summarization, we use two summarization datasets. The MultiLing 2015 dataset [15] is a collection of 15 document sets of news articles from the WikiNews website. Each document set contains 10 news texts about the same event such as 2005 London bombings or the 2004 tsunami. The task is to provide a single fluent summary of 250 words maximum.

The second dataset is DUC 2002 single document summarization dataset⁴. In

⁴ *<https://duc.nist.gov/duc2002/>

our experiment, we use random 100 news articles and produce system summaries that not exceed 100 words. For each document , a human made summary is provided as a reference.

4.3 Summarization system

We use a multilingual summarizer [11] to generate extractive summaries. The summarizer first performs sentence clustering to identify main topics within source texts. Second, terms are ranked according to their relevance to each topic using minimum Redundancy and Maximum Relevance feature selection algorithm [16]. Finally, a score is assigned to each sentence according to the terms mRMR scores. The system summary keeps top relevant sentences up to the summary maximum size.

Top relevant sentences could be extracted from different source documents and paragraphs which necessarily affects summaries coherence. Finding a better ordering of output sentences will improve summary’s coherence

4.4 Genetic algorithm parameters

In addition to fitness function, there is a set of parameters that should be fixed such as crossover and mutation probability, population size and number of generations. For our experimentations, we have fixed population size at 300 individuals, the number of generations at 300, mutation probability at 0.001 and crossover probability at 0.01.

We deliberately decrease the crossover probability since crossover operator generated invalid individuals (summaries that contain duplicate sentences or exceed desired size).

4.5 Evaluation protocol

As described in 1, we define eight configurations for output summary generation: Baseline, thematic ordering and genetic ordering.

Baseline the first configuration represents our baseline: ordering sentences following the original source text ordering. We assume that baseline ordering introduces gaps between sentences since sentences’ sequence is broken.

Topline we consider as a topline, Dicomer scores of reference summaries. Since reference summaries are human made, we assume that it is an upper bound for Dicomer coherence scores.

Rule this configuration combines between our baseline (original ordering) and thematic ordering (see pseudo algorithm 1). Sentences follow first thematic ordering and within each topic, sentences are ordered following their positions.

Coherence model ordering we define several configurations according to the number of positif/negatif input features and the number of sentences to be considered as an input. Here, shared entities feature is combined with thematic ordering, sentence position in the fitness function. Sentence relevance and redundancy penalty features are considered when the model take as an input sentences that exceed the size limit (125% and 150% in our configurations). Then, the model selects a subset of sentences that optimize fitness function score with respect to summary size.

Table 1: Configurations for output summaries orderings

Baseline	SUMBA	[TopN, Position]
Topline	SUMMA	Model summary A MultiLing 2015
Topline	SUMMB	Model summary B MultiLing 2015
Topline	SUMMC	Model summary C MultiLing 2015
Topline	SUMMD	Model summary C DUC 2002
Rule	SUMTP	[Thematic, Position]
Genetic	SUMG1	[+Entity,+Thematic,+Position]
Genetic	SUMG2	[+Thematic]
Genetic	SUMG3	[+Entity]
Genetic	SUMG4	[+Entity,+Thematic]
Genetic	SUMG5	[125%, +Entity,+Thematic,+Position,+Relevance,- Redundancy]
Genetic	SUMG6	[150%, +Entity,+Thematic,+Position,+Relevance,- Redundancy]

4.6 Results and discussion

Figures 2 and 3 report Dicomer coherence scores for each configuration. Topline (Human reference summaries) coherence scores reaches an upper bound of 1.9 for MultiLing 2015 dataset and 1.87 for DUC 2002 dataset.

Baseline system summaries following original orderings (SUMBA) coherence scores is 1.41 for Multiling dataset and 1.29 for DUC 2002 Dataset. Thematic ordering combined with shared entity (SUMG2,SUMG4) present best coherence score for system summaries for both DUC 2002 dataset with a value of 1.34 and Multiling dataset with a value of 1.59. It is the maximum coherence value of system summaries. However, coherence model scores are average and range from 1.27 when five features are considered (SUMG5, SUMG6) to a value of 1.38 when shared entities are considered along with thematic ordering and sentence position feature for the Multiling dataset (SUMG1).

Baseline coherence scores are particularly high compared to other configuration results. When we examine output summaries of the TopN configuration, we find that TopN sentences are similar (contain most relevant terms) leading to some degree of topical coherence.

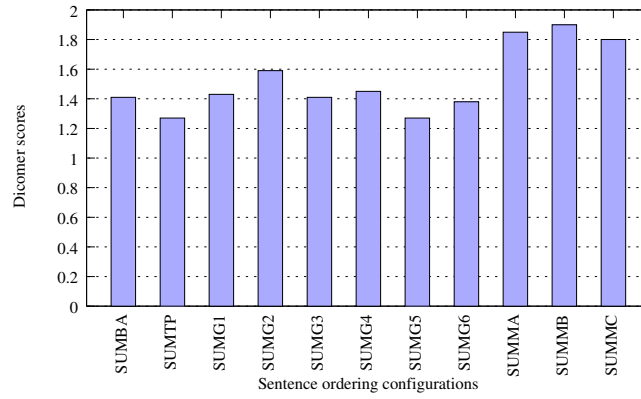


Fig. 2: MultiLing 2015 Dicomer coherence scores

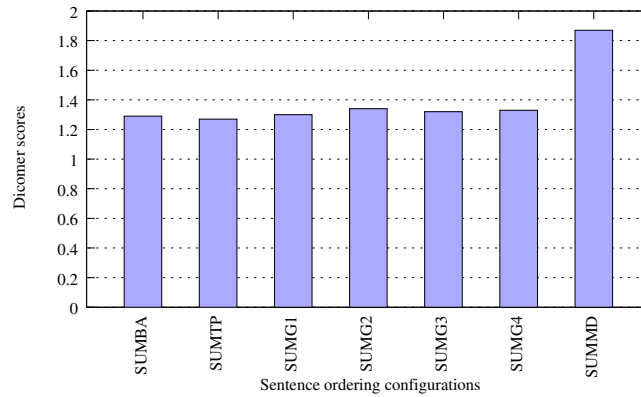


Fig. 3: DUC 2002 Dicomer coherence scores

5 Conclusion

Dealing with text coherence is a challenging task in the NLP field. Taking into account coherence is critical to design efficient tools for text generation which is essential to a range of NLP tasks such as automatic summarization, dialog systems and machine translation. Modeling coherence involves syntactic and semantic levels of discourse analysis: entity-transition patterns, thematic ordering and rhetorical discourse relations. The difficulty with is in defining coherence features and operating all its aspects in a single model.

In this work, we have defined a first model of coherence which combines features that, we assume, have positive/negative input and enhance/affect text coherence. We have designed a genetic algorithm model that take into account a set of coherence features: shared entities, thematic ordering, sentence position, relevance and redundancy. The last three features are useful for target task: extractive summarization. We have experimented different combinations of features thanks to the flexibility of the model and its ability to easily include/exclude features. Due to the nature of source texts (news texts which contains significant amount of date phrases), the results are strongly affected by the dissolution of temporal sequences. Temporal relations are also an important aspect of global coherence and should be considered for future experimentations [17]. Another possible interesting direction is to make the model task independent. Some features that we have defined, such as sentence position and relevance, are task-related and could not be considered for other NLP tasks.

References

1. Slakta, D.: L'ordre du texte (The Order of the Text). *Etudes de Linguistique Appliquee* **19**, 30–42 (1975)
2. Barzilay, R.: The Rhetorical Parsing of Natural Language Texts. *Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics, EACL '97*, 96–103 (1997)
3. Barzilay, R., Lapata, M.: Modeling Local Coherence: An Entity-Based Approach. In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, ACL'05*, 25–30 (2005)
4. Barzilay, R., Elhadad, N., McKeown, K.R.: Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research* **1**(17), 35–55 (2002)
5. Knight, K., Marcu, D.: Summarization Beyond Sentence Extraction: A Probabilistic Approach to Sentence Compression. *Journal of Artificial Intelligence* **139**(1), 91–107 (2002)
6. Molina, A., Torres-Moreno, J., SanJuan, E., da Cunha, I., Martinez, G. E. : Discursive sentence compression. In: *International conference on Computational Linguistics and Intelligent Text Processing*, pp. 394–407. Springer, Samos, Greece (2013)
7. Christensen, J., Soderland, S., Etzioni, O.: Towards Coherent Multi-Document Summarization. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, HLT-NAACL'2013*, 1163–1173 (2013)

8. Li, J., Hovy, E.: A Model of Coherence Based on Distributed Sentence Representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP'2014, 2039–2048 (2014)
9. Nguyen, D. T., Joty, S.: A Model of Coherence Based on Distributed Sentence Representation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL'2017, 13201330 (2017)
10. Charolles, M.: Introduction aux problèmes de la cohérence des textes: Approche théorique et étude des pratiques pédagogiques. *Langue française* **1**(38), 7–41 (1978)
11. Oufaida, H., Blache, P., Nouali, O.: Using Distributed Word Representations and mRMR Discriminant Analysis for Multilingual Text Summarization. In: Biemann, C., Handschuh, S., Freitas, A., Meziane, F., Mtais, E. (eds.) *Natural Language Processing and Information Systems 2015, LNCS*, vol. 9103, pp. 51–63. Springer, Heidelberg (2015).
12. Razali, N. M., Geraghty, J.: Genetic algorithm performance with different selection strategies in solving TSP. In: Proceedings of the world congress on engineering, 1–6 (2011)
13. Fortin, F., Rainville, D., Gardner, M., Parizeau, M, Gagn, C.: DEAP : Evolutionary algorithms made easy. *Journal of Machine Learning Research* **13**(1), 2171–2175 (2012)
14. Lin, Z., Liu, C., Ng, H. T., Kan, M. Y.: Combining coherence models and machine translation evaluation metrics for summarization evaluation. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, ACL'12, 1006–1014 (2012)
15. Giannakopoulos, G., Kubina, J., Conroy, J., Steinberger, J., Favre, B., Kabadjov, M., Kruschwitz, U., Poesio, M.: MultiLing 2015: Multilingual Summarization of Single and Multi-Documents, On-line Fora, and Call-center Conversations. In: Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGDIAL'15, 270–274 (2015)
16. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence Journal*, 1226–1238 (2005)
17. Muller, P., Tannier, X.: Annotating and Measuring Temporal Relations in Texts. In: Proceedings of the 20th International Conference on Computational Linguistics, COLING'04, P.50 (2004)