algorithms

IMPACT FACTOR 2.3

CITESCORE 3.7

Article

# A System to Support Readers in Automatically Acquiring Complete Summarized Information on an Event from Different Sources

Pietro Dell'Oglio, Alessandro Bondielli and Francesco Marcelloni

Special Issue

Machine Learning in Social Network Analytics

Edited by

Dr. Mukesh Prasad, Dr. Faezeh Karimi, Prof. Dr. Dinesh Vishwakarma and Dr. Zahid Akhtar

MDPI

https://doi.org/10.3390/a16110513

*algorithms*

MDPI

*Article*

# A System to Support Readers in Automatically Acquiring Complete Summarized Information on an Event from Different Sources

**Pietro Dell'Oglio [1], Alessandro Bondielli [2] and Francesco Marcelloni [3,*]**

[1] Department of Information Engineering, University of Florence, Via di S. Marta, 3, 50039 Florence, Italy; pietro.delloglio@unifi.it
[2] Department of Computer Science, University of Pisa, Largo Bruno Pontecorvo, 3, 56127 Pisa, Italy; alessandro.bondielli@unipi.it
[3] Department of Information Engineering, University of Pisa, Largo Lucio Lazzarino, 1, 56122 Pisa, Italy
* Correspondence: francesco.marcelloni@unipi.it

**Abstract:** Today, most newspapers utilize social media to disseminate news. On the one hand, this results in an overload of related articles for social media users. On the other hand, since social media tends to form echo chambers around their users, different opinions and information may be hidden. Enabling users to access different information (possibly outside of their echo chambers, without the burden of reading entire articles, often containing redundant information) may be a step forward in allowing them to form their own opinions. To address this challenge, we propose a system that integrates Transformer neural models and text summarization models along with decision rules. Given a reference article already read by the user, our system first collects articles related to the same topic from a configurable number of different sources. Then, it identifies and summarizes the information that differs from the reference article and outputs the summary to the user. The core of the system is the sentence classification algorithm, which classifies sentences in the collected articles into three classes based on similarity with the reference article: sentences classified as dissimilar are summarized by using a pre-trained abstractive summarization model. We evaluated the proposed system in two steps. First, we assessed its effectiveness in identifying content differences between the reference article and the related articles by using human judgments obtained through crowdsourcing as ground truth. We obtained an average F1 score of 0.772 against average F1 scores of 0.797 and 0.676 achieved by two state-of-the-art approaches based, respectively, on model tuning and prompt tuning, which require an appropriate tuning phase and, therefore, greater computational effort. Second, we asked a sample of people to evaluate how well the summary generated by the system represents the information that is not present in the article read by the user. The results are extremely encouraging. Finally, we present a use case.

**Keywords:** natural language processing; text similarity; Transformers; neural language models; newspaper articles



**Citation:** Dell'Oglio, P.; Bondielli, A.; Marcelloni, F. A System to Support Readers in Automatically Acquiring Complete Summarized Information on an Event from Different Sources. *Algorithms* **2023**, *16*, 513. https://doi.org/10.3390/a16110513

Academic Editors: Mukesh Prasad, Faezeh Karimi, Dinesh Vishwakarma and Zahid Akhtar

Received: 10 October 2023
Revised: 3 November 2023
Accepted: 4 November 2023
Published: 8 November 2023



## 1. Introduction

Today, many people use social media platforms to acquire information or to spread information around the world. From this perspective, social media platforms are popular as hubs for the collection and enjoyment of news. Thus, social media has become a useful tool for newspapers, which use social media platforms to disseminate news and attract users. These users, in turn, are inundated with a vast array of information, often interpreted from different personal and political perspectives.

Newspapers differ from each other in the way they deal with specific topics. It can be difficult for a user to obtain clean and complete information. Different sources could treat the same event in different ways, and some content information could be contradictory.

Furthermore, certain newspapers could spread misleading information or fake news [1]. Thus, each news item can contain, in practice, different details on a specific topic. Users seek to gather as much information as possible on a topic to have the opportunity to form their personal opinions. However, they would like to avoid reading all related news items, which often contain redundant information, choosing instead to only focus on content differences.

Starting from this observation, we propose a system that takes a newspaper article, denoted as the *reference article*, and retrieves a set of articles, denoted as *target articles*, on the same topic; the system automatically assesses the contents of these articles in order to pinpoint similarities and differences with respect to the reference article by means of sentence-level similarity. The identified differences can be interpreted as novel and possibly interesting content. Our end goal is to provide the users with an abstractive summarization of such content, thus reducing their reading time and effort.

The system is based on a methodology that leverages the similarity between sentences to find groups of them with varying degrees of similarity. In order to define and exploit a metric for determining sentence similarity, we turn to the literature on natural language processing (NLP), particularly distributional semantics. Most modern approaches in this area are based on neural language models and, more recently, Transformer architectures. At their core, such models define a data-driven probability distribution over sequences of tokens. A token is typically an approximation of the concept of a word and a discrete entity in texts. This distribution is used to learn a fixed-length representation of textual units in a latent space. Distances between such representations in the space can be used to approximate semantic similarity by exploiting the so-called distributional hypothesis [2], which states that similarities in distribution tend to have similar meanings. Thus, similar representations reflect similar word meanings. In recent years, Transformer architectures such as BERT [3] have established themselves as the de facto standard, obtaining state-of-the-art results in many NLP-related tasks, including token-level contextualized representations. In the context of sentence-level comparisons, we can identify two major shortcomings in employing a standard classification approach. First, as the authors of BERT pointed out, sequence-level representations are not to be considered as semantically relevant for determining similarity between sentences [3]. Indeed, token-level representations extracted with BERT-like models tend to obtain results that are generally on par with—if not better than—more traditional approaches, such as word2vec [4] for token-level semantic similarity. Nevertheless, sequence-level representations are typically learned as feature vectors for subsequent classification tasks, thus BERT-like models are not very reliable for determining similarity between sentences. Second, employing a classification model for such a problem may prove to be a rather costly approach. Transformer-based neural language models typically require GPU acceleration during the training phase.

As for the quality of sentence-level representations, we considered Sentence-BERT [5]. It is a method used for further fine-tuning a Transformer model that leverages Siamese and triplet network structures, semantic textual similarity, and natural language inferencing training tasks in order to improve the sequence-level representations provided by a Transformer model. Sentence-BERT is specifically designed to derive semantically meaningful sentence embeddings that can be compared using cosine as a similarity metric [5]. This latter aspect is helpful when designing approaches aimed at reducing the computational cost of solving the problem. We can argue that the computational cost of Transformer-based models is a slightly less prominent issue if we consider them only from a feature extraction perspective. Thus, we propose leveraging Sentence-BERT to obtain sentence-level similarity ratings for newspaper articles.

We exploit these similarities for the sentence classification task described in Section 3. Specifically, we leverage the cosine similarity between sentences as a proxy of their similarity to classify sentences of the target articles with one of three classes, namely *similar* (SS), *different* (DS), and *very different* (VDS), representing the difference level with sentences in the reference article, based on their cosine similarity. Although our aim is to discriminate between similar and different sentences, we realized that some sentences are neither too

similar nor too different. Thus, we introduce an intermediate class that represents sentences that are different, but not distinct enough to be certain that they contain new information with respect to the reference article. Then, we consider all the target articles and employ a similar approach in order to further refine our sentence selection. Specifically, we aim to discard sentences in the target articles that are similar to the sentences in other target articles, in order to reduce the number of overall sentences considered and avoid duplicated information. We describe this process in Section 3.3. Clearly, semantic differences lie on a spectrum and are challenging to reduce to distinct classes. However, we believe that choosing a three-class distinction allows us to avoid performing a too-simplistic distinction between similar and non-similar sentences while giving us enough descriptive power to perform a reliable early analysis of the results. Finally, we employ a Transformer-based abstractive summarization model in order to generate easy-to-read summaries of novel information based on the survived sentences. We describe this process in Section 3.4.

We performed a set of experiments aimed at evaluating both the performances of the sentence classification and the quality of the summarization model for our goal. More specifically, we measured the effectiveness of the sentence classification in identifying novel content in target articles, with respect to the reference article by comparison with human judgments. Such judgments were obtained by means of crowdsourcing. This also enabled us to release a dataset of sentences labeled for similarity, which is likely to be very helpful in several tasks. In addition to this, we compared the performance of our system against a BERT-based classifier that was fine-tuned for the same task, and we employed the same BERT model as a frozen model, applying the prompt tuning technique to utilize it to solve the task. We show that our system is able to achieve comparable performance at a fraction of the computational cost. In fact, while the standard classification pipeline is based on (i) fine-tuning the model, or learning soft prompts to condition the frozen model to the classification task, and (ii) producing the results at inference time by feeding all the test data to the trained classifier, our pipeline allows us, on the one hand, to leverage a pre-trained model without additional tuning on the data itself and, on the other hand, to provide the final results by simply considering the similarity matrix between sentences. This is much more efficient to compute, thus making our system more efficient. Concerning the generated summaries, we again performed a crowdsourcing experiment aimed at evaluating how the summary is perceived by human readers in comparison with the reference article and with the sentences used to generate it. In particular, we aimed to assess, for each summary, the novelty with respect to the reference article, and the completeness and factual correctness of information with respect to the sentences that generated it.

The paper is organized as follows. In Section 2, we present an overview of the state of the art in the field of semantic textual similarity and some related studies that concern textual differences. Section 3 provides an overview of the system and describes in detail the single modules. Section 4 shows some experiments performed to validate the sentence classification process. Finally, in Section 5, we draw some conclusions and highlight possible future directions.

## 2. Related Works

Semantic textual similarity is a challenging task in NLP and text mining and is closely related to the field of distributional semantics. From the linguistic perspective, distributional semantics is based on a simple assumption called *distributional hypothesis*. For the distributional hypothesis, the more two words are semantically similar to each other, the more they tend to appear in the same, or similar, linguistic context due to the fact that "difference of meaning correlates with difference of distribution" [6]. From a computational perspective, we can suppose that words are represented as vectors encoding the properties of their contexts in a vector space. Their (semantic) similarity is given by the distance between their vector representations. We usually refer to vector representations of words as word embeddings [2].

The NLP field has made large use of the distributional hypothesis and the distributional properties of words to encode their meaning. While the earliest attempts exploited co-occurrence matrices to represent words based on their contexts, more modern approaches leverage machine learning and deep learning in the form of neural language models. Among such models, the earliest ones typically employed unsupervised learning to obtain fixed-length representations of words [4] and sentences [7].

In recent years, Transformer-based neural language models have established themselves as the de facto standard for many NLP tasks. BERT is an architecture based on the self-attention mechanism to deal with complex tasks involving human language [3]. A very attractive aspect of the BERT-like architectures is that their internal representations of words and sequences are context-aware. The attention mechanism in Transformers facilitates the consideration of relationships between words within a sentence or across more significant portions of a text, establishing deep connections. Furthermore, researchers have proposed other architectures with attention-based mechanisms, such as AlBERT [8] and DistilBERT [9], which have gained significant attention and continue to be exploited by the NLP community. Nevertheless, it is fair to admit that BERT and similar models face some limitations, especially when applied to tasks related to semantic textual similarity, particularly at the level of sentence-level embeddings.

One of the limitations of BERT and BERT-like models is evident in tasks regarding semantic textual similarity, particularly when coping with sequence-level embeddings [5]. It is well known that BERT's sequence-level embeddings are not directly trained to encode the semantics of the sequences and, thus, are not suited to compare them with standard metrics such as cosine similarity [3]. To overcome these limitations, Sentence-BERT [5] was proposed. Sentence-BERT is a modification of the pre-trained BERT network with Siamese and triplet network structures. It can produce sentence embeddings that are semantically meaningful and compared using a similarity measure (for example, cosine similarity or Manhattan/Euclidean distance). The process of finding the most similar pair is reduced from 65 h with BERT/RoBERTa to about 5 s with Sentence-BERT while maintaining the accuracy achieved by BERT [5]. Research has been conducted to design and evaluate various approaches for employing Siamese networks, similarity concepts, one-shot learning, and context/memory awareness in textual data [10]. Furthermore, recent efforts have focused on developing an unsupervised contrastive learning method that transforms pre-trained language models into universal text encoders, as seen with Mirror-BERT [11] and subsequent models [12].

In recent years, we have also seen the rise of large language models, with a considerable number of parameters reaching the tens or even hundreds of billions. These models differ from their predecessors in terms of scale and, in some instances, they incorporate reinforcement learning techniques during training. Prominent examples of large language models include OpenAI's GPT [13], Google's LLaMA [14], and Hugging Face's BLOOM [15]. These models typically outperform smaller counterparts and are recognized by their zero-shot learning capabilities and emergent new abilities [16]. However, they are affected by two significant limitations. First, most of these models are controlled by private companies and are only accessible via APIs. Second, the computational costs of such models often pose challenges for running them on standard commercial hardware without resorting to parameter selection and/or distillation techniques.

The problem of semantic textual similarity between pairs of sentences has been discussed in several papers and some studies have faced the issue of extracting semantic differences from texts. In particular, research has been carried out on the ideological discourses in newspaper texts. For instance, some authors proposed a statistical model for ideological discourse based on the assumption that we can detect ideological perspectives through lexical variants [17]. Following this idea, others investigated the utility of applying text-mining techniques to support the discourse analysis of news reports. They found contrast patterns to highlight ideological differences between local and international press coverage [18]. In recent years, critical discourse analysis has been applied to investigate ide-

ological differences in reporting the same news across various topics, such as the COVID-19 in Iranian and American newspapers [19], and the representation of Syrian refugees in Turkey, considering three Turkish newspapers [20].

Recent research in the realm of newspaper text analysis has explored a variety of topics. Sentiment analysis, for instance, has been a key area of focus, as seen in reference [21–23], and more recently in [24]. Some approaches, such as reference [25], and resources, such as the SentiCoref 1.0 dataset [26], have also been made available in recent years. Another related topic is the application of opinion mining techniques on noisy data, such as blogs. A semi-automated approach (in which these texts are first cleaned using domain knowledge and then subjected to mining) was proposed in [27]. Moreover, studies have been conducted on the narrative and factual evolution of news articles. A novel dataset on news revision histories, named NewsEdits, and three novel tasks aimed at predicting actions performed during article version updates have been made available in [28]. Here, the authors show that, to date, these tasks are possible for expert humans but are challenging for large NLP models. Furthermore, the application of statistical learning algorithms to pattern analysis not only in the newspaper domain but also in different media has been studied in [29]. Several studies have been conducted in completely different domains, such as scholarly documents. A hybrid model, which considers both section headers and body text to recognize automatically generic sections in scholarly documents, was proposed in [30]. There also exist studies in the corpus linguistics domain that provide quantitative evidence by examining collocations [31]. These studies typically use frequency, keywords, or concordance-based pattern analysis techniques.

Our system uses an abstractive text summarization model. These kinds of models produce a paraphrasing of the main contents of the given text, using a vocabulary set different from the original document. Several recent models in abstractive text summarization exist. Some of these models are based on classical neural network architectures with attention mechanisms [32] and conditional RNN [33]. Other abstractive text summarization models are based on an encoder–decoder architecture with attention [34]. The CopyNet model incorporates the copying mechanism in the neural encoder–decoder model for sequence learning tasks [35]. More recent studies applied the Transformer architecture to neural abstractive summarization [36]. Bidirectional and autoregressive Transformers (BART) [37] and text-to-text transfer Transformer (T5) [38] are two widely used sequence-to-sequence pre-trained models. The model we used for our system was PEGASUS (pretraining using extracted gap sentences for abstractive summarization by sequence-to-sequence models) [39].

It is also important to note systems with approaches or applications similar to ours. For instance, ReviewChomp https://www.reviewchomp.com/ (accessed on 3 November 2023) is based on multi-document text summarization and provides customer review summaries for various products or services. Users can search for the desired product or service on the website. It is essential to note that while multi-document text summarization and our system share some similarities, they have different objectives. Our system focuses on extracting information differences from several target articles based on a reference article already read by the user. In contrast, multi-document text summarization approaches primarily aim to summarize information extracted from multiple texts. One interesting and recent application is the *lecture summarization service*, which is a Python-based RESTful service that utilizes a BERT model for text embedding and K-means clustering to identify the sentences closest to the centroid for summary selection. The primary aim of this service is to equip students with a tool capable of generating summaries of lectures in a specified number of sentences [40]. A service offering similar summary generation for lengthy lecture videos is detailed in [41].

## 3. The Proposed System

The flowchart of the proposed system is presented in Figure 1. The system expects a reference article as input. Then, it retrieves a set of target articles that refer to the same

topic/event from a set of sources (e.g., newspaper websites) and outputs a summary of the information, which is contained in the target articles and is considered to not be very similar to the one contained in the reference article. We attempt to capture the desires of the typical news readers who read an article from their favorite newspaper websites but also wish to acquire opinions on the same event from other newspapers, possibly of different political connotations. The system consists of four main steps: target article retrieval, sentence classification, target sentence reduction, and target sentence summarization. The core of the system is the algorithm used in the 'Sentence Classification' module, which allows labeling the sentences of the target articles into three classes, namely similar sentence (SS), different sentence (DS), and very different sentence (VDS),based on the values of similarity with the reference article computed exploiting sentence embeddings extracted by a Sentence-BERT pre-trained model. We will detail this algorithm in Section 3.2.

Before performing sentence classification, several pre-processing operations are applied to the articles. In particular, we perform standard text annotation actions, such as sentence splitting and named entity recognition. We also extract keywords using KeyBERT. Then, we use a Sentence-BERT model to obtain the representation for each sentence. Finally, we perform the target sentence classification: each sentence is associated with one of three similarity classes.

Concerning the target sentence reduction, we filter out all the target sentences that contain redundant information with respect to the target articles (i.e., those sentences classified as different from all the reference sentences, but containing information redundant with other target articles). This aspect is described in detail in Section 3.3.

Finally, for the target sentence summarization, we first choose the sentences to be shown to the user from each target article. Then, we exploit an abstractive text summarization algorithm to generate a summary of the text resulting from the union of these sentences in order to provide an easy-to-read output for the user. We detail the target sentence summarization in Section 3.4.

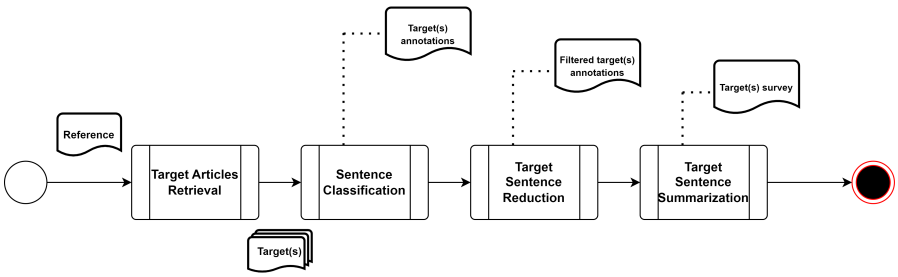


**Figure 1.** Flowchart of the proposed system.

### *3.1. 'Target Articles Retrieval'*

The first module of our pipeline is concerned with retrieving news articles on the same topic or event as the reference article. As previously mentioned, we denote the retrieved articles as target articles. Given a reference article, already read by the user, the module collects articles from a pre-defined set of sources. In the current implementation, sources are widely popular newspaper websites. Each source homepage is scraped in order to retrieve target articles referring to the same topic or event reported in the reference article. In order to assess the degree of similarity of the article acquired by scraping with the reference article, we exploit a simple yet effective and efficient method based on named entity recognition (NER). Specifically, named entities are first extracted from the reference and target articles using the spaCy NLP pipeline spaCy.io (accessed on 3 November 2023) spaCy is an open-source text processing library in Python. The version utilized is 3.5.0, and the module exploited is the `en_core_web_lg`. As for spaCy, we chose it for its widespread use, both in industry and academia, and its ease of implementation in the pipeline. As for the specific `en_core_web_lg` model, we used it for two main reasons. First, the model has been trained on web-based data and, thus, is expected to be effective for our task. Second, our goal was to achieve a good balance between the accuracy of the model and its computational cost.

Although a more accurate Transformer-based model exists for the English language, it is more expensive to run locally and requires GPU acceleration. The chosen model allows the module to easily run on a CPU-only system while achieving good accuracy.

Articles are considered as target articles if they contain at least a third of the named entities extracted from the reference article. We chose this threshold after an empirical evaluation, guided starting with the following consideration. The use of named entities provides reasonable confidence in retrieving articles whose topics align with that of the reference article, as they refer to the same named entities. Simultaneously, selecting articles that do not share all named entities with the reference article, but only some, ensures that the target articles contain new or different information with respect to the reference article.

This step allows us to efficiently retrieve possible target articles from the web without performing in-depth similarity calculations, which is a more expensive task. Thus, we are able to reduce the number of articles that have to be compared with the reference article, with negligible loss of information. The system will calculate the similarity between the target and reference articles in the subsequent steps, involving more comprehensive analysis and specific techniques.

Note that the set of sources can be redefined provided an appropriate functionality for retrieving information from the new sources. In addition to this, the threshold could be considered as a parameter in future upgraded versions of the application.

### *3.2. Sentence Classification*

The 'Sentence Classification' module is based on the algorithm described in the pseudo-code shown in Algorithm 1. Upon receiving a reference article and the previously described set of target articles as input, the 'Sentence Classification' module categorizes all sentences within the target articles as SS, DS, or VDS.

First, the reference article is split into sentences by using the spaCy pipeline. Furthermore, keywords from the reference article are extracted by using KeyBERT https://github.com/MaartenGr/KeyBERT (accessed on 3 November 2023). These keywords are merged with the named entities of the reference article extracted by the 'Target Articles Retrieval' module to generate a list of representative words for the reference article. KeyBERT leverages Sentence-BERT to obtain word embeddings. Then, it selects keywords under the assumption that their cosine similarity is the highest with respect to the embedding of the whole document.

Second, with the aim of obtaining reliable representations of semantically relevant news, we exploit a Sentence-BERT pre-trained model to encode the sentences of the reference article. More specifically, we employ the `distiluse-base-multilingual-cased-v2` pre-trained model [42]. It is a multilingual model, and the training is based on the idea that a translated sentence should be mapped to the same location in the vector space as the original sentence. We employed this model to facilitate the generalization of the proposed methodology with newspaper texts in other languages [42].

The sequence of elaborations performed on the reference article is applied to each target article. Once the sentences and their distributional representations with Sentence-BERT are obtained for each target article, the sentences are classified with respect to the similarity with sentences in the reference article by calling the *TargetSentenceLabeling()* function. We formally define our classification problem as follows.

Let $r$ and $T = \{t_1, t_2, \ldots, t_n\}$ be, respectively, a reference news article and a set of one or more target articles. Both $r$ and $t_i$ are represented as sets of sentences, $r = \{s_1^r, s_2^r, \ldots, s_n^r\}$ and $t_i = \left\{s_1^{t_i}, s_2^{t_i}, \ldots, s_{m_i}^{t_i}\right\}$. The goal is to classify each sentence $s_j^{t_i}$ in each $t_i$ with one of the three classes (SS, DS, VDS) based on degrees of similarity with all the sentences in $r$. In particular, the label is assigned to $s_j^{t_i}$ by considering the maximum similarity between $s_j^{t_i}$ and all the sentences in $r$. We employ two thresholds, namely high similarity (HS) and low similarity (LS), to discriminate, respectively, between SS and DS, and DS and VDS. The threshold values are considered as parameters. If at least one sentence in $r$ has a similarity score higher than HS with $s_j^{t_i}$, then $s_j^{t_i}$ is classified into the SS class. If no sentence in $r$ has a

similarity score higher than LS with $s_j^{t_i}$, then $s_j^{t_i}$ is classified into the VDS class. Finally, if none of the two previous conditions are met (i.e., the maximum similarity score between $s_j^{t_i}$ and a sentence in $r$ is between LS and HS thresholds), $s_j^{t_i}$ is classified into the DS class.

**Algorithm 1** The pseudocode of the algorithm used in the 'Sentence Classification' module

*RA*: the reference article (i.e., the article already read by the user)
*TAs*: the list of target articles (i.e., the articles retrieved by the 'Target Articles Retrieval' module. They refer to the same topic or event as the reference article)
*RA_named_entities*: the named entities of the reference article (extracted during the 'Target Articles Retrieval' sub-process)
*TAs_named_entities*: the named entities of the target articles (extracted during the 'Target Articles Retrieval' sub-process)
*S*: the list of target sentences labeled by the algorithm. Each member of the list is a tuple consisting of the target sentence, the label, and the most similar reference sentence to the target sentence.

**function** SENTENCECLASSIFICATION(*RA*, *TAs*, *RA_named_ents*, *TAs_named_ents*, *S*)
    *RA_sent_set* ← *Sentence_splitting*(*RA*) //(the reference article is split into sentences)
    *RA_enc_sent_set* ← ∅
    *S* ← ∅
    *RA_keywords* ← *KeyBERT*(*RA_sent_set*) //(keywords are extracted from the reference article)
    *list_of_RA_repr_w* ← *Merge*(*RA_keywords*, *RA_named_ents*) //(keywords and named entities are merged into a unique list of representative words for the reference article)
    **for** each *sent* in *RA_sent_set* **do**
        *RA_enc_sent_set* ← *Sentence_BERT*(*sent*) //(the sentences of the reference article are encoded by using Sentence-Bert)
    **end for**
    **for** each *target_article* in *TAs* **do**
        *TA_sent_set* ← *Sentence_splitting*(*target_article*) //(the target article is split into sentences)
        *TA_keywords* ← *KeyBERT*(*TA_sent_set*) //(keywords are extracted from the target article)
        *list_of_TA_repr_w* ← *Merge*(*TA_keywords*, *TA_named_ents*) //(keywords and named entities are merged into a unique list of representative words for the target article)
        *TA_enc_sent_set* ← ∅
        **for** each *sent* in *TA_sent_set* **do**
            *TA_enc_sent_set* ← *Sentence_BERT*(*sent*) //(the sentences of the target article are encoded by using Sentence-Bert)
        **end for**
        *S* ← *TargetSentenceLabeling*(*RA_enc_sent_set*, *TA_enc_sent_set*, *S*)
        *S* ← *LengthBasedRefinement*(*S*, *list_of_TA_repr_w*, *list_of_RA_repr_w*)
        *S* ← *KeywordBasedRefinement*(*S*, *list_of_TA_repr_w*, *list_of_RA_repr_w*)
        **if** all the sentences of the target article are labeled as VDS **then**
            remove all the sentences from S
        **end if**
    **end for**
    **return** *S*
**end function**

Once each sentence in the target article has been classified, we tune our labeling by taking the sentence length into account by calling the *LengthBasedRefinement()* function.

Finally, we further tune the labeling by considering shared keywords between sentences by calling the *KeywordBasedRefinement()* function. If all the sentences in the target article are labeled as VDS, the article is removed (all its sentences are removed). This enables us to integrate the check based on the named entities executed in Section 3.1 with an additional check based on cosine similarity, in order to delete potential articles that do not align with the topic of the reference article.

Notably, we can highlight some important differences between the proposed system and typical approaches to sentence similarity. These approaches usually compare pairs of sentences to obtain a similarity score, e.g., a value in the [0, 1] range or a similarity label. These sentences are often considered out of their context. Our system takes into account sentence-level similarity in a document-level context: each target sentence (and, thus, each target article) is evaluated based on its similarity with a specific set of reference sentences. The three sentence-level classes provide us with a way to determine document-level similarity as well, answering the question "how many and which parts of the target document are similar/dissimilar to the reference one?".

In the following, we detail the three functions used in the Sentence Classification module.

#### 3.2.1. Target Sentence Labeling

Function *TargetSentenceLabeling()* is described in the pseudo-code shown in Algorithm 2. Once each sentence $s_j^{t_i}$ in $t_i$ and each sentence $s_l^r$ in $r$ are represented as embeddings, we compute the similarity between $s_j^{t_i}$ and $s_l^r$, for each sentence in $t_i$ and $r$, by exploiting the cosine similarity. In Table 1, we summarize how the sentences $s_j^{t_i}$ in $t_i$ are classified based on values of maximum cosine similarity and provide a description for each class.

**Table 1.** Description of the three classes and corresponding classification rules based on the maximum cosine similarity between the sentence $s_j^{t_i}$ of the target article $t_i$ under analysis and the sentences of the reference article $r$.

| Labels | Classification Rules | Description |
|---|---|---|
| SS | **IF** maximum cosine similarity between $s_j^{t_i}$ and each sentence in reference article $r \geq$ HS **THEN** classify $s_j^{t_i}$ as SS | Sentences in this class include pieces of information that are shared with the reference article. This information likely refers to the objective description of the event. |
| DS | **IF** maximum cosine similarity between $s_j^{t_i}$ and each sentence in reference article $r <$ HS **AND** $\geq$ LS **THEN** classify $s_j^{t_i}$ as DS | Most of the sentences in this class include pieces of information that are likely shared with reference sentences but reported differently. |
| VDS | **IF** maximum cosine similarity between $s_j^{t_i}$ and each sentence in reference article $r <$ LS **THEN** classify $s_j^{t_i}$ as VDS | Sentences containing information different from the information in the reference sentences; it also contains some noise. |

**Algorithm 2** The pseudocode of the TargetSentenceLabeling() function

```
function TargetSentenceLabeling(RA_enc_sent_set, TA_enc_sent_set, S)
    for each TAsent in TA_enc_sent_set do
        if Max_Cosine_Similarity(TAsent, RA_enc_sent_set) ≥ HS then
            label ←' SS'
        else if Max_Cosine_Similarity(TAsent, RA_enc_sent_set) < LS then
            label ←' VDS'
        else
            label ←' DS'
        end if
        RAsent ← Most_Similar_to_Target(RA_enc_sent_set)
        TAtuple ← [TAsent, label, RAsent]
        S.append(TAtuple)
    end for
    return S
end function
```

#### 3.2.2. Length-Based Refinement

The annotation step described in the previous subsection suffers from the following problem: when sentences vary significantly in length, their calculated similarity may be low despite them sharing similar content. Indeed, length is an essential factor for BERT-like models. Although sentence-BERT has proved to provide semantically meaningful representations of sentences in the vector space [5], the cosine similarity computed between embeddings of sentences with similar content but different lengths is not generally as high as expected for the scope of the presented system. Table 2 shows an illustrative example. The two sentences express the same information. Nevertheless, since their difference in length is high, the cosine similarity computed between the corresponding embeddings is too small for labeling the target sentence as SS. Thus, the system reiterates all target–reference sentence pairs, specifically focusing on those target sentences not labeled as SS, and that are significantly shorter in terms of tokens compared to the reference sentence (specifically, the reference sentence is at least two times longer in terms of tokens with respect to the target one). The system then determines whether the longer sentence contains additional information by employing the following method, which is based on the function *LengthBasedRefinement()* described in the pseudo-code shown in Algorithm 3.

**Table 2.** Examples of sentences that share pieces of core information but with a low cosine score, equal to 0.58.

| Source | Sentence |
|---|---|
| Fox News (target) | *'The FBI also has been called to investigate the incident.'* |
| CNN International (reference) | *'The FBI is investigating the incident*, which drew widespread condemnation of the officers after a video showing part of the encounter circulated on social media.' |

**Algorithm 3** The pseudocode of the LengthBasedRefinement() function

```
function LengthBasedRefinement(S, list_of_TA_repr_w, list_of_RA_repr_w)
    for each TAtuple in S do
        if TAtuple.label ≠' SS' then
            if len(TAtuple.RAsent) ≥ 2 * len(TAtuple.TAsent) then
                TA_sent_repr_w ← Get_Sent_W(TAtuple.TAsent, list_of_TA_repr_w)
                RA_sent_repr_w ← Get_Sent_W(TAtuple.RAsent, list_of_RA_repr_w)
                list_of_shared_w ← Get_Shared_W(TA_sent_repr_w, RA_sent_repr_w)
            end if
            list_ref_frag ← []
            for each shared_w in list_of_shared_w do
                list_ref_frag.append(left_frag(TAtuple.RAsent, shared_w))
                list_ref_frag.append(right_frag(TAtuple.RAsent, shared_w))
            end for
        end if
        if Max_Cosine_Similarity(TAtuple.TAsent, list_reference_frag) ≥ HS then
            TAtuple.label ←' SS'
        end if
    end for
    return S
end function
```

The system considers the most representative words of the reference sentence shared with the target one. They consist of the keywords extracted using KeyBERT and named entities. Then, the system uses the left and right contexts for each keyword and named entity in the reference sentence to obtain a set of text fragments.

Each text fragment is compared with the target sentence. If the two sentences share some content information, then at least one (*target sentence*, *text fragment*) pair will obtain a cosine greater than or equal to HS. If this occurs, the system changes the labeling from VDS or DS to SS for the target sentence; otherwise, no change is made. The iteration continues until there are no more pairs in the VDS or DS groups to compare.

As an example, Table 3 shows a fragment obtained from the reference sentence of CNN International in Table 2. This fragment is compared to the target sentence of Fox News, with a cosine score of 0.8.

**Table 3.** An example of a fragment–reference sentence pair with a high cosine similarity score. Shared keyword is highlighted in bold.

| Fragment | Target Sentence | Cosine Similarity |
|---|---|---|
| 'The **FBI** is investigating the incident' | 'The **FBI** also has been called to investigate the incident.' | 0.8 |

#### 3.2.3. Keyword-Based Refinement

Another aspect we have to take into account is the fact that the Sentence-BERT representations often position sentences that share numerous significant words (e.g., named entities, keywords) closer together in the latent space, thus hindering the descriptive power of the cosine metric in these cases. Table 4 provides an example.

**Table 4.** Two sentences with a high cosine score that report different information about the same topic but share a set of keywords and named entities (highlighted in bold).

| Reference Sentence | Target Sentence | Cosine Similarity |
|---|---|---|
| 'Pfizer anticipates applying for emergency use authorization for a **third dose** of its vaccine as soon as next month, **Mikael Dolsten**, who leads worldwide research, development and medical for **Pfizer**, was quoted by CNN as saying at the teleconference.' | 'During a company earnings call on Wednesday morning, Dr. **Mikael Dolsten**, who leads worldwide research, development and medical for **Pfizer**, called the new data on a **third dose** of vaccine ëncouraging.' | 0.7 |

Here, sentences share a set of named entities (Mikael Dolsten, Pfizer, third dose), and this yields a high cosine similarity. However, their informative content (and arguably, meaning) is rather different. To deal with this limitation, we further re-examine all the target sentences included in the SS and DS groups. For each of these sentences, the system checks if the information contained in the target sentence is similar to the one included in the reference sentence, excluding all the common keywords and named entities, by adopting the following method, which is based on the function *KeywordBasedRefinement()* described in the pseudo-code shown in Algorithm 4.

For each pair $s_j^{t_i}$ in $t_i$ and $s_l^r$ in $r$, first, the common keywords and named entities are identified. Then the system generates text fragments from the two sentences, excluding those keywords and named entities. Finally, the system compares each text fragment from the reference article with all the text fragments from the target article. If all the combinations of text fragments extracted have a cosine score lower than LS, the class of the target sentence is changed to VDS.

At the end of all these steps, the system outputs a set of classified target articles $T_C = \{t_1, t_2, \ldots, t_n\}$, where $t_i = \left\{s_1^{t_i}, s_2^{t_i}, \ldots, s_{n_i}^{t_i}\right\}$ is the set of all the sentences of the article $t_i$, labeled as SS, DS, and VDS.

**Algorithm 4** The pseudocode of the KeywordBasedRefinement() function

```
function KeywordBasedRefinement(S, list_of_TA_repr_w, list_of_RA_repr_wo)
    for each TAtuple in S do
        if TAtuple ≠' VDS' then
            TA_sent_repr_w ← Get_Sent_W(TAtuple.TAsent, list_of_TA_repr_w)
            RA_sent_repr_w ← Get_Sent_W(TAtuple.RAsent, list_of_RA_repr_w)
            list_of_shared_w ← Get_Shared_W(TA_sent_repr_w, RA_sent_repr_w)
            list_ref_frag ← []
            list_target_frag ← []
            for each shared_w in list_of_shared_w do
                list_ref_frag.append(left_frag(TAtuple.RAsent, shared_w))
                list_ref_frag.append(right_frag(TAtuple.RAsent, shared_w))
                list_target_frag.append(left_frag(TAtuple.TAsent, shared_w))
                list_target_frag.append(right_frag(TAtuple.TAsent, shared_w))
            end for
            if Max_Cosine_Similarity(list_target_frag, list_ref_frag) <' LS' then
                TAtuple.label ←' VDS'
            end if
        end if
    end for
    return S
end function
```

### *3.3. Target Sentence Reduction*

Once the labeled set of sentences included in $T_C$ is obtained, we further refine the sentence selection process by determining sentences in $T_C$ that are similar to each other, i.e., redundant. As these sentences generally provide very little further information, we remove them. The target sentence reduction sub-process is based on the function *TargetSentenceReduction()* described in the pseudo-code shown in Algorithm 5.

**Algorithm 5** The pseudocode of the TargetSentenceReduction() function

```
function TargetSentenceReduction(S)
    list_DS_VDS ← []
    for TAtuple in S do
        if TAtuple.label ≠' SS' then
            list_DS_VDS.append(TAtuple)
        end if
    end for
    list_DS_VDS_ranked ← Ranking(list_DS_VDS) //(Each sentence in the target
article is ranked based on the similarity value with sentences in the reference article)
    list_of_filtered_sentences = []
    for TAtuple in list_DS_VDS_ranked do
        if Max_Cosine_Sim(TAtuple, NextSentsInList(TA_tuple, list_DS_VDS_ranked))
< HS then
            list_of_filtered_sentences.append(TAtuple)
        end if
    end for
    return list_of_filtered_sentences
end function
```

First, the system takes the set of labeled target articles $T_C = \{t_1, t_2, \ldots, t_n\}$ as input, and merges the sentences of target articles that are labeled as DS and VDS. SS is disregarded as it can be considered redundant with the reference article. We rank the list of DS plus VDS sentences with respect to their maximum similarity with sentences in $r$ in descending order. Then, starting from the top, for each sentence, we compute its similarity score with

all the other sentences on the list. If at least one score is higher than the HS threshold, i.e., there is another sentence similar to the one considered that also has a lower similarity score with the reference article, we consider the sentence redundant and remove it from the list. The final list includes only sentences that are considered non-redundant.

### *3.4. Target Sentence Summarization*

The goal of the system is to supply users with information they may have missed with the article of their choice (i.e., the reference article) in an easy-to-read format. With the aim of making the information promptly available to the user, the system generates a summary of the target sentences and outputs this summary.

According to our definition, all the sentences labeled VDS are selected as they certainly correspond to sentences with information different from the reference article. In addition, all the sentences classified DS with a cosine similarity score lower than LS; all the sentences in VDS are selected as well. All the sentences selected are merged into a unique text.

The system then creates a summary of this text by exploiting a neural text summarization model. In particular, we use Google PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive Summarization) [39]. It is a pre-trained large Transformer-based encoder–decoder model. PEGASUS uses a gap-sentence generation strategy, in which important sentences are masked by an input document and are then generated together as one output sequence from the remaining sentences.

The final output is a unique and readable text that only reports information included in VDS sentences and some of the DS sentences for each target article. In this way, we can visualize the new content in the target articles with respect to the reference one.

In Table 5, an example derived from article segments on the Notre Dame fire is presented. The reference article is part of an article published on 16 April 2019 on NBC News, while the two target articles are extracted from the set of articles published on the same date by USA Today and Al Jazeera English. These three example articles are included in a dataset of international newspapers (e.g., CNN, The Guardian, Al Jazeera English) about events that happened occurred 2019 and 2021. The dataset is available on GitHub https://github.com/pietrodelloglio/dataset-for-system-acquiring-content-differences (accessed on 3 November 2023).

**Table 5.** Reference article and summary extracted from VDS sentences and some of the DS sentences collected from two target articles about the Notre Dame fire in 2019.

| Reference Article (NBC News) | Summary of Target Articles (USA Today & Al Jazeera English) |
|---|---|
| 'Some of the most prized, centuries-old relics of France and Christianity survived the devastating Notre Dame Cathedral fire that almost wiped out the cherished Paris landmark, authorities said Tuesday. Culture Minister Franck Riester breathed a huge sigh of relief, telling reporters outside Notre Dame that the nation's "most precious treasures" were largely spared. Saved treasures from the Notre Dame Cathedral are temporarily stored in city hall after a massive fire devastated large parts of the Gothic cathedral in Paris. Benoit Tessier/Reuters Many of the works will be stored at Paris' City Hall and the Louvre, where they will be examined, treated for damage, and protected, officials said. Audrey Azoulay, director-general of UNESCO, the U.N. culture agency, said Notre Dame has "a particular place in the world's collective imagination" and has pledged her agency's help to rebuild. Monday's fire almost destroyed the entire cathedral, which has stood in Paris and survived nearly 900 years of tumultuous French history. Paris prosecutor Remy Heitz has launched an investigation, which he said would be "long and complex."' | 'The huge fire that tore through Notre Dame Cathedral in Paris last night has left the famous landmark in ruins, but not as badly as many had feared. The famous spire was reduced to ash, but the building's stone face and bell towers were saved, the BBC reports. The main altar has been badly damaged, but the famous white marble Piet' sculpture by French sculptor Nicolas Coustou is still standing, seemingly unscathed, at its place before the main altar. "The great tragedy is the structure itself and its fittings", a Brown University art professor tells the Los Angeles Times. "Many people don't know that the roofs are timber, largely replaced in the 19th century, and they have been totally consumed. Masonry will burn and degrade under intense heat, which is clearly what is happening, but we can only wait and see what kind of damage has spread to the church within." A full investigation is underway.' |

## 4. Experimental Results

The effectiveness of the proposed approach was evaluated along two dimensions. First, we assessed the performance of the 'Sentence Classification' module in determining a class of similarity between sentences. In particular, we compared the output of the module with human judgment collected by crowdsourcing and with two BERT classifiers. Second, we evaluated how much the summary produced by our approach represents a coherent, factually correct, and complete summarization of the target sentences different from the reference article. Again, we used human judgment collected by crowdsourcing.

For the experiments, we first collected and annotated a dataset of online newspaper articles spanning various topics that serve both as ground truth for performance evaluation and as a training dataset for setting the optimal parameters, especially concerning the thresholds. The dataset simulates a scenario with a reference article and at least two other target articles for a given news item. Target sentences are labeled through crowdsourcing with three classes—SS, DS, and VDS—for their similarity with the sentences in the reference article. We detail the creation of the dataset in Section 4.1. We tuned the model's parameters (i.e., the similarity thresholds) and applied the sentence classification to target articles. We evaluated the performances of the 'Sentence Classification' module in two ways. First, we measured the F1 score with respect to human judgment. Second, we compared the proposed system with two BERT classifiers, applying model tuning and prompt tuning respectively, in order to obtain a reliable comparison with state-of-the-art approaches. Section 4.2 provides insights into the experiment for parameter tuning and a comparison with the two BERT-based classifiers.

Concerning the evaluation of the summaries, we again performed a crowdsourcing experiment, asking participants to, given a reference article and the target sentences, rate the quality of the summary in terms of completeness, factual correctness, and novelty. We detail this experiment in Section 4.3.

Finally, in Section 4.4, we present some examples of how our system works in practice.

### *4.1. Dataset*

We compiled a dataset that includes 43 articles, each relating to one of 8 different news events, with at least 3 articles for each news event. We assume that articles pertaining to the same event and published on the same date (or similar) are comparable. For each news event, we consider one article as the reference article, and the others as target articles.

For each article, we record the news it is sharing, the main topic of that news event (e.g., politics, health, etc.), the newspaper source, the date of publication, and the URL.

To simplify the collection process, we exploited CrowdTangle https://apps.crowdtangle.com/ (accessed on 3 November 2023). We collected a list of events that occurred between 2020 and 2021 and obtained social media posts from Facebook pages of international newspapers (e.g., CNN, The Guardian, Al Jazeera English) with links to articles about the news. We then collected the news content directly from the newspaper website.

In order to obtain ground-truth labels for sentences in the target articles, we performed a crowdsourcing experiment using Prolific https://www.prolific.co/ (accessed on 3 November 2023). The annotators were presented with a list of ⟨target sentence–reference sentence⟩ pairs. Annotators were asked to decide whether the target sentence was similar, different, or very different from the reference sentence. Each pair was labeled by seven different annotators, and the majority vote determined the gold standard label. If the majority vote was uncertain, the pair was discarded.

The dataset is available on GitHub https://github.com/pietrodelloglio/dataset-for-system-acquiring-content-differences (accessed on 3 November 2023).

### *4.2. Target Sentence Classification Evaluation*

First, we aimed to identify the optimal threshold parameters for our proposed system. To achieve this, we conducted a series of experiments involving different values

for LS and HS, and compared the automatic classifications generated by our system with human labeling.

We employed a leave-one-out cross-validation approach, where one news article was held out as a test set, while the remaining articles were used to determine the threshold values that best aligned with human labeling. For each fold, we obtained eight different LS and HS pairs that achieved the best performance according to the highest number of news articles. In case of a tie, we randomly selected one pair. We obtained the same LS and HS pairs for all folds. This suggests that the choice of LS and HS thresholds was not particularly sensitive.

Table 6 shows an example of F1 scores obtained in an example fold. The table refers to the following events: "Mars Landing" (M.L.), "Italy Lockdown" (I.L.), "Italy Euro 2020" (I.E.), "Global Warming" (G.W.), "George Floyd Death" (G.F.), "Elliot Page coming out" (E.P.), and "China Landed on Mars" (C.L.).

**Table 6.** F1 scores of different threshold configurations in an example fold.

| LS | HS | M.L. | I.L. | I.E. | G.W. | G.F. | E.P. | C.L. |
|---|---|---|---|---|---|---|---|---|
| 0.4 | 0.6 | 0.46 | 0.61 | 0.62 | 0.61 | 0.47 | 0.87 | 0.59 |
| 0.4 | 0.7 | 0.52 | 0.58 | 0.63 | 0.62 | 0.47 | **0.89** | 0.61 |
| 0.4 | 0.8 | 0.54 | 0.58 | 0.65 | 0.62 | 0.47 | **0.89** | 0.59 |
| 0.5 | 0.6 | 0.6 | 0.75 | 0.75 | 0.7 | **0.79** | 0.85 | 0.75 |
| 0.5 | 0.7 | 0.65 | **0.79** | 0.76 | 0.71 | **0.79** | 0.87 | 0.77 |
| 0.5 | 0.8 | **0.67** | **0.79** | 0.78 | **0.72** | **0.79** | 0.87 | 0.75 |
| 0.6 | 0.7 | 0.56 | 0.72 | 0.82 | 0.71 | **0.79** | 0.87 | **0.88** |
| 0.6 | 0.8 | 0.58 | 0.72 | **0.84** | **0.72** | **0.79** | 0.87 | 0.86 |

We selected LS = 0.5 and HS = 0.8, as these values are the best parameters in the highest number of news events.

Table 7 displays the F1 score performance of the test news for each fold. On average, our system achieved an F1 score of 0.77, with the lowest value being 0.67 and the highest reaching 0.87.

**Table 7.** F1 score for the single news and the average F1 score.

| News | F1 score |
|---|---|
| Mars Landing | 0.67 |
| Italy Lockdown | 0.79 |
| Italy Euro 2020 | 0.78 |
| Global Warming | 0.72 |
| George Floyd Death | 0.79 |
| Elliot Page | 0.87 |
| China Landed on Mars | 0.75 |
| Beirut Explosion | 0.81 |
| **Average** | **0.772** |

To compare our system with state-of-the-art classification models, we utilized two BERT classifiers based on the `bert-base-uncased` model [3]. In the first comparison, we fine-tuned the model within the leave-one-out cross-validation setting. The fine-tuning parameters were set as follows: a maximum sequence length of 128, a learning rate of $2 \times 10^{-5}$, and a batch size of 8 (due to computational constraints). The model was trained for 5 epochs in each fold, with the remaining parameters following the standard settings of the Hugging Face implementation https://huggingface.co/bert-base-uncased (accessed on 3 November 2023).

In the second comparison, we employed prompt tuning with the following parameters: a maximum sequence length of 128, a learning rate of $3 \times 10^{-2}$, and a batch size of 8 (again, due to computational limitations). The model was trained for 30 epochs in each fold.

We present the results of the two BERT classifiers and compare them with our system in Table 8. The table demonstrates that, on average, fine-tuned BERT performs slightly better than our system, which, in turn, outperforms the model tuned with prompts. It is worth noting that fine-tuning a classification model based on BERT can be expensive in terms of both time and computational costs. In contrast, our system is significantly more efficient in both the training and classification phases, requiring only the extraction of sentence-level embeddings and the computation of the similarity matrix. Prompt tuning, on the other hand, tends to work better with larger models and demands substantial computational resources.

**Table 8.** F1 score for the single news event and the average F1 score for our classifier and two BERT classifiers.

| News | Our System | Fine-Tuned BERT | Prompt-Tuned BERT |
|---|---|---|---|
| Mars Landing | 0.67 | 0.73 | 0.56 |
| Italy Lockdown | 0.79 | 0.75 | 0.61 |
| Italy Euro 2020 | 0.78 | 0.82 | 0.77 |
| Global Warming | 0.72 | 0.72 | 0.66 |
| George Floyd Death | 0.79 | 0.74 | 0.58 |
| Elliot Page | 0.84 | 0.92 | 0.83 |
| China Landed on Mars | 0.75 | 0.89 | 0.84 |
| Beirut Explosion | 0.81 | 0.81 | 0.56 |
| **Average** | **0.772** | **0.797** | **0.676** |

We computed the Wilcoxon signed-rank test to verify if the differences between the performances of our system and the two BERT classifiers are statistically significant. For the computation of the statistics, we used the SciPy method https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wilcoxon.html (accessed on 3 November 2023), maintaining the default parameters. In the case of the comparison between our system and the fine-tuned BERT, we obtained $W = 5.0$ (the minimum of the sum of ranks above and below zero) and $pvalue = 0.87$. Thus, we can conclude that there is no statistically significant difference between the two classifiers. In the case of the comparison between our system and the prompt-tuned BERT, we obtained $W = 4.5$ and $pvalue = 0.05$. Thus, we can conclude that there is a statistically significant difference between the two classifiers.

In Figures 2–7, we provide examples of confusion matrices for two of the analyzed news articles, respectively, the coming out of Elliot Page and Italy's victory in the Euro 2020 Championship.

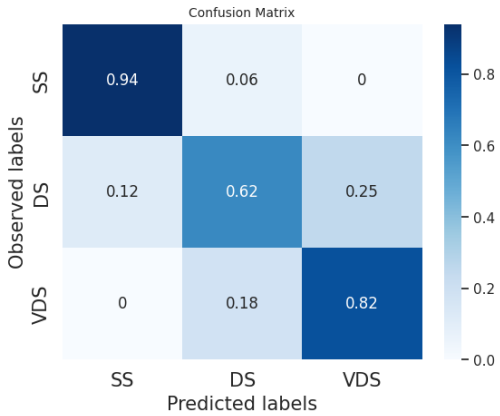


**Figure 2.** Confusion matrix of our system on the news about Elliot Page coming out.

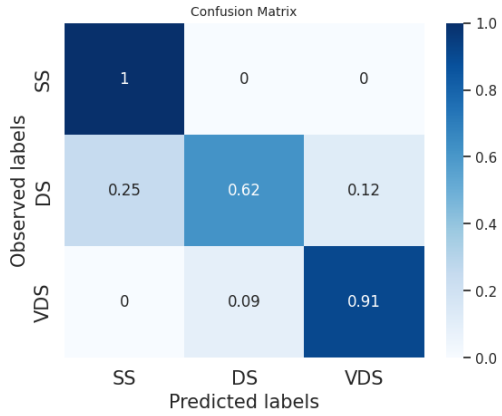


**Figure 3.** Confusion matrix of the fine-tuned BERT on the news about Elliot Page coming out.

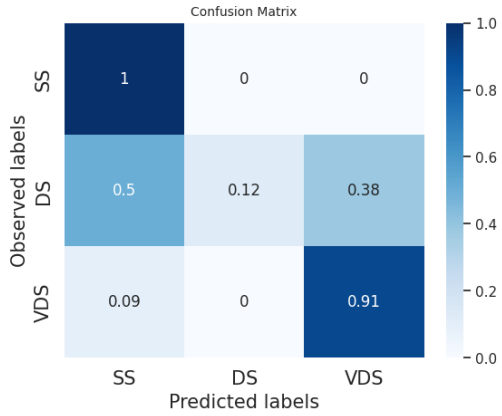


**Figure 4.** Confusion matrix of the prompt-tuned BERT on the news about Elliot Page coming out.

The analysis of the figures shows that the results are dependent on the class and the news item. For example, our system and the fine-tuned BERT perform equally on the DS class in the news about Elliot Page coming out, while our system obtains better performances than the fine-tuned BERT on the same class in the case of the news about Italy's victory, even if the fine-tuned BERT has a higher F1 score in general.

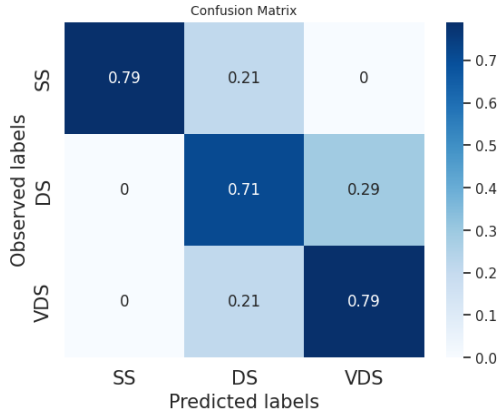


**Figure 5.** Confusion matrix of our system on the news about Italy winning Euro 2020.

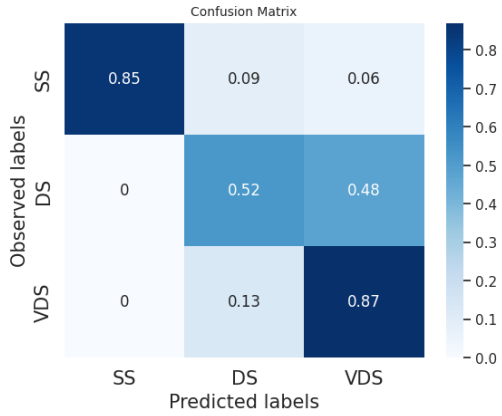


**Figure 6.** Confusion matrix of the fine-tuned BERT on the news about Italy winning Euro 2020.

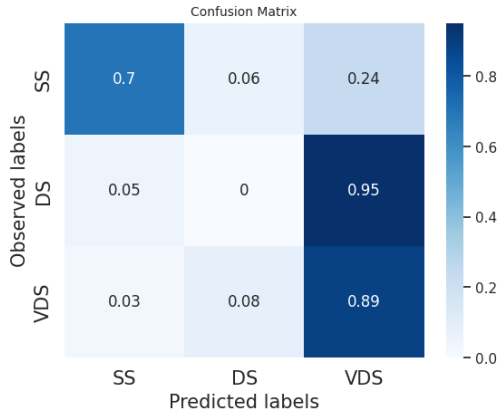


**Figure 7.** Confusion matrix of the prompt-tuned BERT on the news about Italy winning Euro 2020.

Figures 8–10 show the average of the confusion matrices on all the tested news. Here, we can observe that the BERT classifiers perform slightly better than our system on the VDS class, while our system performs better on the DS class. Furthermore, the fine-tuned BERT outperforms our system and prompt-tuned BERT on the SS class. In general, the SS and DS classes present the greatest challenges for both BERT classifiers and our system. In the first case, we point out that several events in our datasets are slightly unbalanced with respect to the SS class. This is explained by the fact that two different articles on the same event often have a few similar sentences belonging to the SS class while the majority of the sentences belong to the DS and VDS classes. Conversely, the DS class is certainly the most challenging to predict. This difficulty may arise from the fact that the DS class is the least homogeneous of the three. Typically, it includes sentences with the same informative content as the most similar sentence in the reference article but expressed in a different linguistic style.

We can conclude that the sentence classification method implemented in our system works quite well compared to the labeling performed by human users and used as ground truth. Furthermore, on average, our classifier achieves slightly lower performance compared to the BERT classifier trained with model tuning. However, it is important to note that BERT always requires fine-tuning with annotated data, whereas our system can be used without any tuning phase. Indeed, we can directly employ default thresholds, as demonstrated in our experiments.

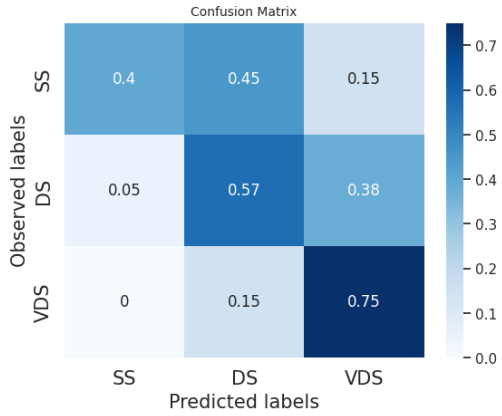


**Figure 8.** Average confusion matrix of our system.

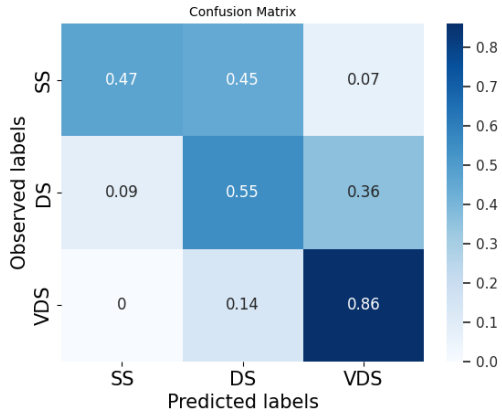


**Figure 9.** Average confusion matrix of the fine-tuned BERT.

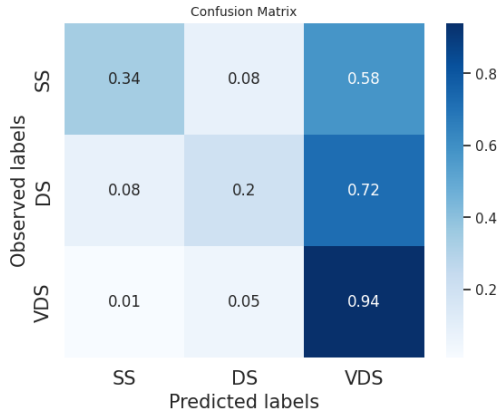


**Figure 10.** Average confusion matrix of the prompt-tuned BERT.

### *4.3. Text Summarization Evaluation*

In order to evaluate the quality of the generated summaries, we resorted again to crowdsourcing. We carried out an experiment to evaluate the quality of the information contained in the summaries generated from the target sentences extracted from our system. Specifically, for each of the news items in the dataset, we presented the raters with (i) the reference article; (ii) the target sentences extracted from the target articles, as described in Sections 3.2 and 3.3; and (iii) the summary generated from such sentences, as described in Section 3.4. For each summary, raters were asked to provide a rating for (i) the novelty, with respect to the reference article; (ii) factual correctness, with respect to the target sentences; and (iii) information completeness, with respect to the target sentences. Each summary

was rated on these 3 characteristics by 7 different people on a Likert-type scale from 1 to 7, with 1 and 7 being the lowest and the highest scores, respectively.

Figure 11 shows the human judgments for each news event regarding the novelty, i.e., the amount of new information contained in the target summary compared to the reference article. On average, the score is 3.5. Some news items have better evaluations than others. In particular, the summaries on Elliot Page coming out, George Floyd's death, and the Mars landing obtain majority scores between 4 and 6, while the news about Italy winning the Euro championship obtains a low score.

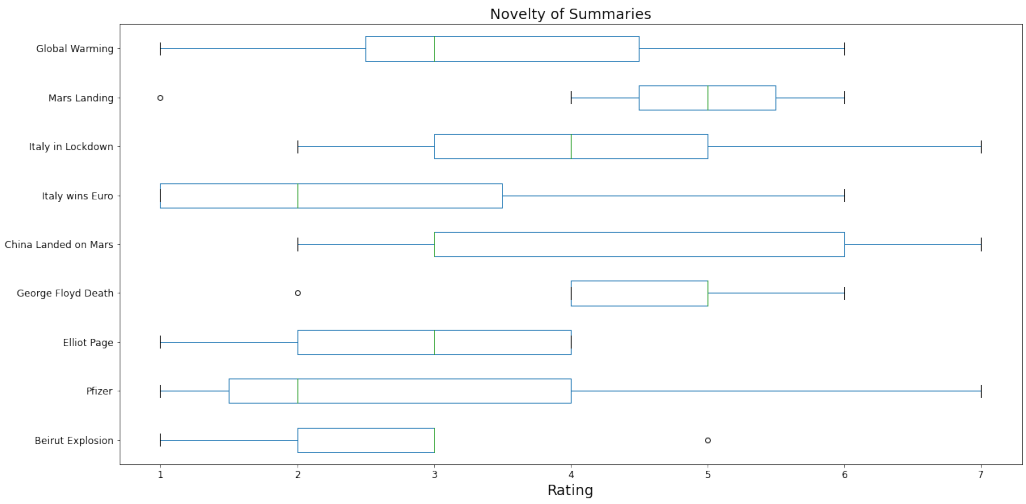


**Figure 11.** Box plot of human judgments for each news event on the amount of new information contained in the summary, with respect to the reference article.

Figure 12 presents a box plot depicting the human judgments for each news event regarding the factual correctness of the target summary in comparison to the reference article. The average score is 4.6, which is quite high, with the majority of scores for nearly all news items ranging between 5 and 6.

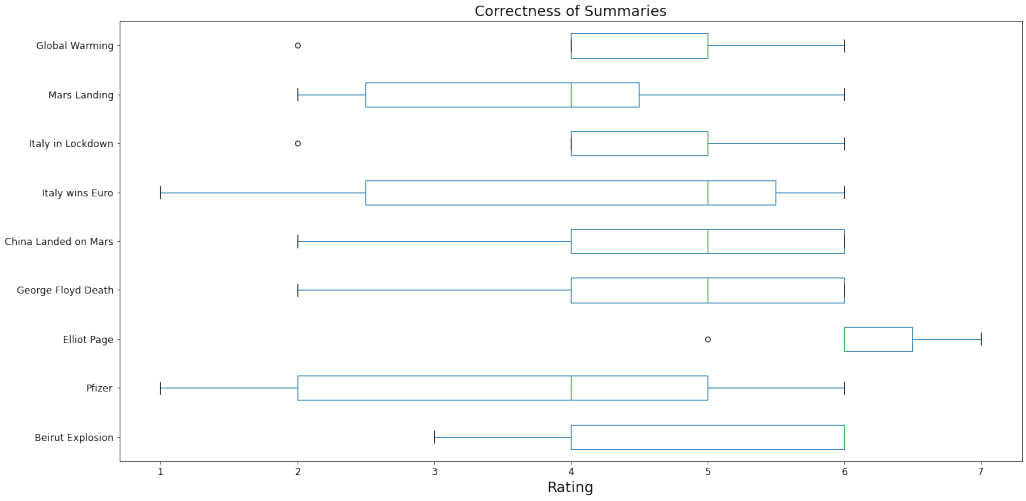


**Figure 12.** Box plot depicting human judgments on the factual correctness of target summaries for each news eventrelative to the reference article.

Finally, Figure 13 presents human judgments on information completeness for each news event, with an average score of 3.8, which is encouraging, as we observe that the majority of scores are high for certain news events, such as Elliot Page coming out and George Floyd's death, while they are slightly lower for others.

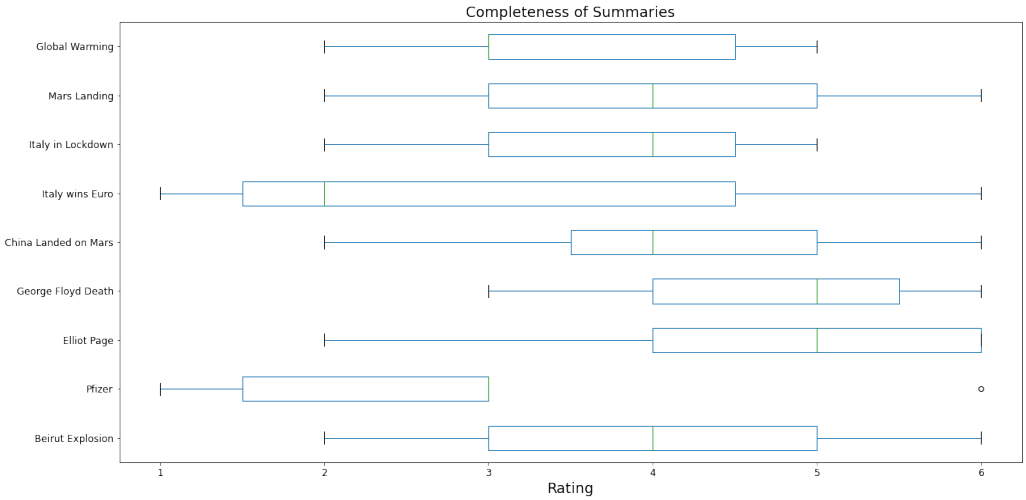


**Figure 13.** Box plot of human judgments for each news event, assessing the information completeness in the summaries of target sentences extracted from the target articles, in comparison to the reference article.

The experiment shows that the automatically generated summaries are able to provide an adequate overview of the target sentences. Specifically, correctness ratings are, on average, rather satisfactory, indicating that the summaries provide factually correct texts with respect to the original news. Completeness ratings are also quite high, even though a significant drop in quality is observed for some of the considered news. Novelty ratings, while generally lower, are still encouraging. Thus, we can conclude that, although some users are less satisfied than others, depending on the specific news, on average, the system is able to output comprehensible and coherent summaries.

### *4.4. An Example of the Application*

In this section, we show an example of how the proposed system can be applied. We choose a reference news article about the war in Ukraine, titled "Second victim of New Year's Eve strikes on Kyiv dies in hospital", published on CNN International on 2 January 2023. The system was run on the same date at 10:21 a.m. (CET). The text and the link to the article are included in Attachment S1 of the supplemental material. The 'Target Articles Retrieval' module collected four different articles, all on the same event but focusing on different aspects. The newspaper websites selected for the search were The New York Times, NBC News, Al Jazeera English, USA Today, and BBC News. The system retrieved articles only from NBC News, Al Jazeera English, and USA Today websites. On that particular day, there was no article similar to the reference article in the New York Times and on BBC News websites. The titles of the target articles are listed in Table 9.

**Table 9.** The metadata of the target articles.

| Title | Event | Source | Date |
|---|---|---|---|
| Music and missile strikes: Ukraine rings in the New Year under Russian attack. | Ukraine war | NBC News | 2 January 2023 |
| Air raid sirens in Kyiv as Russia launches fresh drone attacks. | Ukraine war | Al Jazeera English | 2 January 2023 |
| Renewed Russian attacks mark Ukraine's grim start to 2023. | Ukraine war | Al Jazeera English | 2 January 2023 |
| Kyiv residents ringing in new year caught in Russian attack; Zelenskyy says Ukraine will not forgive: Updates. | Ukraine war | USA Today | 2 January 2023 |

The second module of the system takes the target and reference articles as input. Then, the articles are split into sentences, as explained in Section 3. First, the system performs an automated classification of all sentences in the target articles (see Section 3.2). Table 10 shows a subset of target sentences from the Al Jazeera English article titled "Renewed Russian attacks mark Ukraine's grim start to 2023", labeled with this approach.

**Table 10.** Partial output of the sentence labeling sub-process.

| Target Sentence | Label | Reference Sentence | Cosine |
|---|---|---|---|
| Elsewhere, a 22-year-old woman died of wounds from a Saturday rocket attack in the eastern town of Khmelnytskyi, the city's mayor said. | DS | Another person died and 20 others were injured in Saturday's explosions, Klitschko said. | 0.5446 |
| While Russia's bombardments have left many Ukrainians without heating and electricity due to damage or controlled blackouts meant to preserve the remaining power supply, Ukraine's. | VDS | A 46-year-old man who was injured by a Russian attack on the Ukrainian capital on Saturday has died in hospital, according to city mayor Vitali Klitschko. | 0.2609 |
| "The power industry is doing everything possible to ensure that the New Year's holiday is with light, without restrictions", utility company Ukrenergo said. | VDS | "One of the injured as a result of the Russian attack on the capital on December 31 died this morning", Klitschko said on his official Telegram channel. | 0.0954 |
| Russia's RIA state news agency cited a local doctor as saying six people were killed when a hospital in Donetsk was attacked on Saturday. | DS | A 46-year-old man who was injured by a Russian attack on the Ukrainian capital on Saturday has died in hospital, according to city mayor Vitali Klitschko. | 0.5216 |

As for the target sentence reduction (Section 3.3), for each target article, all the labeled sentences are filtered to obtain only sentences that are not considered redundant. Given the reference article, the system for each target article outputs the list of target sentences that contain new or different information with respect to the reference article. This output is included in Attachment S2 of the supplemental material. Each row in the CSV file "S2" consists of the target sentence, the labeling, the reference sentence with the highest similarity to the target one, the similarity score, and an integer that represents the index of the target article assigned by the system.

Finally, the output of the target sentence summarization (Section 3.4) is a text summary that reports only novel or different information with respect to the reference article. The sentences included in Attachment S2 are used as input for the abstractive text summarization algorithm. Attachment S3 of the supplemental material includes the results of the target sentence summarization for the example. In particular, we show the summaries of each target article and the summary generated from the sentences of all the target articles. These sentences are those outputted by the target sentence reduction module.

An illustrative simple example is shown in Table 5. The interface of the application is shown in Figure 14.

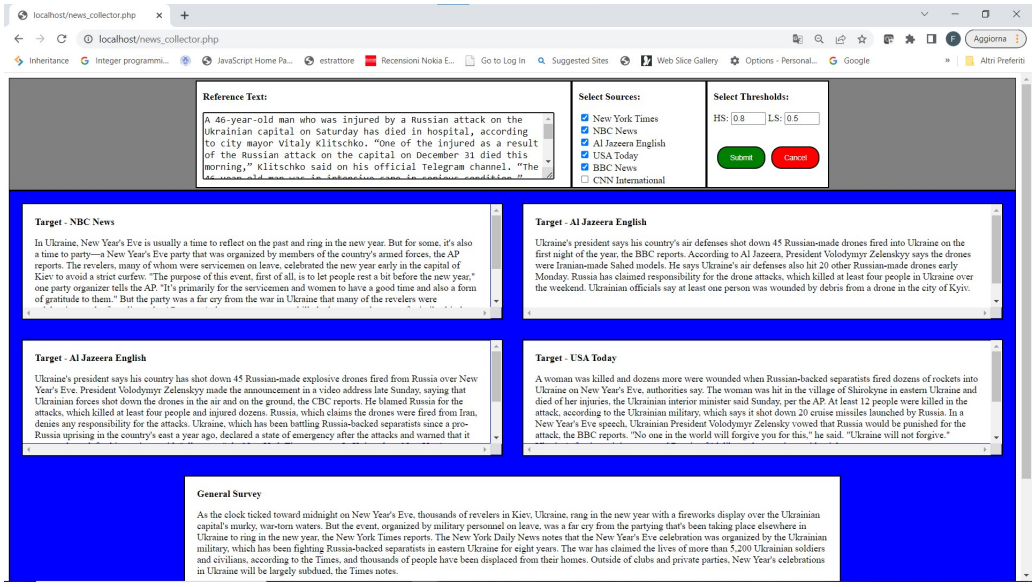


**Figure 14.** Interface of the application.

## 5. Conclusions

In this paper, we proposed a method for (i) identifying novel information from newspaper articles with respect to another article on the same topic, and (ii) making such information available in an easy-to-read format to users. We consider the setting in which there is one reference article (i.e., the article read by a user), and one or more target articles on the same topic. The system employs cosine similarity based on Sentence-BERT embeddings to classify each target sentence as similar, different, or very different, with respect to sentences in the reference article. We implemented a sequence of steps to extract the most relevant sentences (i.e., the different and very different ones) from the target articles. Then, we employed a Transformer-based abstractive summarization model to summarize the sentences in an easy-to-read format.

We conducted experiments using crowdsourced human judgments as the ground truth to assess the classifier's effectiveness and the summary's ability to represent information in the target articles that differs from the reference article. We obtained an average F1 score of 0.772 against average F1 scores of 0.797 and 0.676 achieved by, respectively, a fine-tuned BERT classifier and a prompt-tuned BERT classifier. By computing the Wilcoxon signed-rank test, our classifier was statistically equivalent to the first BERT classifier and statistically outperformed the second one. This is a remarkable result considering that our classifier requires less computational effort.

Regarding the summaries—based on the human judgments collected through crowdsourcing, the summaries have been deemed to accurately represent the information contained in the selected sentences of the target articles

We strongly believe that the large availability of data prevents today's users from finding the information necessary to create their own opinions. The proposed system can be considered a first step toward the end goal of empowering users to discover novel and relevant information, and will be further improved in future work. In particular, we intend to investigate other similarity metrics and classification algorithms to better address the problem of semantic similarity between sentences. Furthermore, we plan to improve the quality of summaries, especially concerning their novelty, by leveraging different methods. Finally, our ultimate goal is to build a platform that allows users to submit a reference article and receive a summary with information different from the reference article.

**Supplementary Materials:** The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/a16110513/s1.

**Author Contributions:** Conceptualization, P.D., A.B., and F.M.; methodology, P.D., A.B., and F.M.; software development, P.D.; validation, P.D.; data preparation, P.D.; writing—original draft preparation, P.D.; writing—review and editing, A.B. and F.M.; visualization, P.D.; supervision, F.M.; project administration, F.M.; funding acquisition, F.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has been partially funded by the PNRR-M4C2-Investimento 1.3, Partenariato Esteso PE00000013—"FAIR-Future Artificial Intelligence Research"—Spoke 1 "Human-centered AI" as part of the NextGenerationEU program, and the Italian Ministry of University and Research (MUR) under the framework of the PRIN 2022JLB83Z "Psychologically-tailored approaches to Debunk Fake News detected automatically by an innovative artificial intelligence approach", the FoReLab and CrossLab projects (Departments of Excellence).

**Data Availability Statement:** The data used in this study are available on GitHub at https://github.com/pietrodelloglio/dataset-for-system-acquiring-content-differences (accessed on 3 November 2023).

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## References

1. Bondielli, A.; Marcelloni, F. A survey on fake news and rumour detection techniques. *Inf. Sci.* **2019**, *497*, 38–55. [CrossRef]
2. Lenci, A. Distributional semantics in linguistic and cognitive research. *Ital. J. Linguist.* **2008**, *20*, 1–31.
3. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
4. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 3111–3119.
5. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Hong Kong, China, 3–7 November 2019; Association for Computational Linguistics: Kerrville, TX, USA, 2019.
6. Harris, Z.S. Distributional structure. *Word* **1954**, *10*, 146–162. [CrossRef]
7. Le, Q.V.; Mikolov, T. Distributed Representations of Sentences and Documents. *arXiv* **2014**, arXiv:1405.4053.
8. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *arXiv* **2019**, arXiv:1909.11942.
9. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
10. Holeček, M. Learning from similarity and information extraction from structured documents. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2021**, *24*, 149–165. [CrossRef]
11. Liu, F.; Vulić, I.; Korhonen, A.; Collier, N. Fast, effective, and self-supervised: Transforming masked language models into universal lexical and sentence encoders. *arXiv* **2021**, arXiv:2104.08027.
12. Liu, F.; Jiao, Y.; Massiah, J.; Yilmaz, E.; Havrylov, S. Trans-Encoder: Unsupervised sentence-pair modelling through self-and mutual-distillations. *arXiv* **2021**, arXiv:2109.13059.
13. OpenAI. GPT-4 Technical Report. *arXiv* **2023**, arXiv:2303.08774.
14. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. Llama: Open and efficient foundation language models. *arXiv* **2023**, arXiv:2302.13971.
15. Scao, T.L.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A.S.; Yvon, F.; Gallé, M.; et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv* **2022**, arXiv:2211.05100.
16. Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; et al. Emergent abilities of large language models. *arXiv* **2022**, arXiv:2206.07682.
17. Lin, W.H.; Xing, E.; Hauptmann, A. A joint topic and perspective model for ideological discourse. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Antwerp, Belgium, 15–19 September 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 17–32.
18. Pollak, S.; Coesemans, R.; Daelemans, W.; Lavrač, N. Detecting contrast patterns in newspaper articles by combining discourse analysis and text mining. *Pragmatics* **2011**, *21*, 647–683. [CrossRef]
19. Dezhkameh, A.; Layegh, N.; Hadidi, Y. A Critical Discourse Analysis of COVID-19 in Iranian and American Newspapers. *GEMA Online J. Lang. Stud.* **2021**, *21*, 231–244.
20. Onay-Coker, D. The representation of Syrian refugees in Turkey: A critical discourse analysis of three newspapers. *Continuum* **2019**, *33*, 369–385. [CrossRef]

21. Balahur, A.; Steinberger, R. Rethinking Sentiment Analysis in the News: From Theory to Practice and back. *Proceeding WOMSA* **2009**, *9*, 1–12.
22. Garvey, C.; Maskal, C. Sentiment analysis of the news media on artificial intelligence does not support claims of negative bias against artificial intelligence. *Omics J. Integr. Biol.* **2020**, *24*, 286–299. [CrossRef]
23. Shrestha, B.B.; Bal, B.K. Named-Entity Based Sentiment Analysis of Nepali News Media Texts. In Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications, Suzhou, China, December 2020; pp. 114–120.
24. Luo, M.; Mu, X. Entity sentiment analysis in the news: A case study based on negative sentiment smoothing model (nssm). *Int. J. Inf. Manag. Data Insights* **2022**, *2*, 100060. [CrossRef]
25. Lin, S.Y.; Kung, Y.C.; Leu, F.Y. Predictive intelligence in harmful news identification by BERT-based ensemble learning model with text sentiment analysis. *Inf. Process. Manag.* **2022**, *59*, 102872. [CrossRef]
26. Žitnik, S.; Blagus, N.; Bajec, M. Target-level sentiment analysis for news articles. *Knowl.-Based Syst.* **2022**, *249*, 108939. [CrossRef]
27. Dey, L.; Haque, S.M. Opinion mining from noisy text data. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2009**, *12*, 205–226. [CrossRef]
28. Spangher, A.; Ren, X.; May, J.; Peng, N. NewsEdits: A News Article Revision Dataset and a Novel Document-Level Reasoning Challenge. In Proceedings of the Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Seattle, WA, USA, 10–15 July 2022; pp. 127–157.
29. Fortuna, B.; Galleguillos, C.; Cristianini, N. Detection of bias in media outlets with statistical learning methods. In *Text Mining*; CRC: Boca Raton, FL, USA, **2009**; pp. 27–50.
30. Li, S.; Wang, Q. A hybrid approach to recognize generic sections in scholarly documents. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2021**, *24*, 339–348. [CrossRef]
31. Baker, P. *Using Corpora in Discourse Analysis*; A&C Black: London, UK, 2006.
32. Rush, A.M.; Chopra, S.; Weston, J. A neural attention model for abstractive sentence summarization. *arXiv* **2015**, arXiv:1509.00685.
33. Chopra, S.; Auli, M.; Rush, A.M. Abstractive sentence summarization with attentive recurrent neural networks. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 93–98.
34. Nallapati, R.; Zhou, B.; Gulcehre, C.; Xiang, B. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv* **2016**, arXiv:1602.06023.
35. Gu, J.; Lu, Z.; Li, H.; Li, V.O. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv* **2016**, arXiv:1603.06393.
36. Gehrmann, S.; Deng, Y.; Rush, A.M. Bottom-up abstractive summarization. *arXiv* **2018**, arXiv:1808.10792.
37. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv* **2019**, arXiv:1910.13461.
38. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 5485–5551.
39. Zhang, J.; Zhao, Y.; Saleh, M.; Liu, P. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 11328–11339.
40. Miller, D. Leveraging BERT for extractive text summarization on lectures. *arXiv* **2019**, arXiv:1906.04165.
41. Srikanth, A.; Umasankar, A.S.; Thanu, S.; Nirmala, S.J. Extractive text summarization using dynamic clustering and co-reference on BERT. In Proceedings of the 2020 5th International Conference on Computing, Communication and Security (ICCCS), Patna, India, 14–16 October 2020; pp. 1–5.
42. Reimers, N.; Gurevych, I. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. In Proceedings of the Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online, 16–20 November 2020.