






Modeling and Summarizing News Events Using Semantic Triples

Radityo Eko Prasajo^(✉) , Mouna Kacimi , and Werner Nutt 

Free University of Bozen-Bolzano, Piazza Domenicani 3, 39100 Bozen-Bolzano, Italy
{RPrasajo, Mouna.Kacimi, Werner.Nutt}@unibz.it

Abstract. Summarizing news articles is becoming crucial for allowing quick and concise access to information about daily events. This task can be challenging when the same event is reported with various levels of detail or is subject to diverse view points. A well established technique in the area of news summarization consists in modeling events as a set of semantic triples. These triples are weighted, mainly based on their frequencies, and then fused to build summaries. Typically, triples are extracted from main clauses, which might lead to information loss. Moreover, some crucial facets of news, such as reasons or consequences, are mostly reported in subordinate clauses and thus they are not properly handled. In this paper, we focus on an existing work that uses a graph structure to model sentences allowing the access to any triple independently from the clause it belongs to. Summary sentences are then generated by taking the top ranked paths that contain many triples and show grammatical correctness. We further provide several improvements to that approach. First, we leverage node degrees for finding the most important triples and facets shared among sentences. Second, we enhance the process of triple fusion by providing more effective similarity measures that exploit entity linking and predicate similarity. We performed extensive experiments using the DUC'04 and DUC'07 datasets showing that our approach outperforms baseline approaches by a large margin in terms of ROUGE and PYRAMID scores.

1 Introduction

A large amount of news articles is published daily to cover important events. The volume of such content can be overwhelming for news readers compromising their ability to get an overall, but concise, picture of what is happening. To solve this issue, multi-document summarization approaches were developed providing a quick access to essential information [4, 9, 10, 12, 13, 19, 22, 26]. The main challenge of summarizing news articles is how to capture the different perspectives, view points, and levels of detail reported for the same event. Once these differences are captured, their inclusion in the summary is far from being trivial. For example, different news platforms reported the following information about the same event:

S1: “President Donald Trump has fired FBI Director James Comey, the White House said in a statement Tuesday evening.” (TIME)

S2: “Donald Trump fired FBI director James Comey in order to stop an investigation which could have potentially ruinous consequences for the administration.” (New York Times)

S3: “Democrats said James Comey was fired because the FBI was investigating alleged links between the Trump campaign and Russia.” (BBC)

S4: “The Trump administration attributed Comey’s dismissal to his handling of the investigation into Democratic nominee Hillary Clinton’s email server.” (CNN)

S5: “Donald Trump could be forced to leave office over the investigations into his administration’s links with Russia.” (INDEPENDENT)

The above sentences put the light on different aspects of the dismissal of the FBI director. They give diverse types of details that we call *facets*, including time, news provenance, and possible reasons for firing the FBI director. Thus, the task of summarization consists in first identifying the main facts and their facets from a set of news articles, and then fusing those facets to have a concise description of each event.

This problem falls into the category of abstractive approaches where sentences of the original text are rephrased to create summaries [1, 3, 5, 6, 8–10, 12, 13, 21, 22, 26, 27]. The main idea of abstractive summarization is to leverage fine-grained fact extraction where a fact can be represented as words or semantic triples of the form $\langle \text{Arg1}; \text{predicate}; \text{Arg2} \rangle$. Most existing approaches rely on similarity between facts to merge information [1, 3, 6, 9, 10, 13, 19, 22, 26]. In the previous example, sentences *S1* and *S2* contain the same fact $\langle \text{Donald Trump}; \text{fired}; \text{FBI director} \rangle$ and therefore they can be fused to summarize three facets of information including the source of the news, the time, and the reason. Typically the similarity between sentences is based on facts belonging to main clauses. Thus, if two sentences contain similar facts in subordinate clauses, their fusion is not easily handled. For example, considering the main clause, sentence *S3* talks about firing the FBI director and sentence *S5* talks about “the end of Trump’s presidency”. By contrast, considering the subordinate clause, they both talk about the “Russian investigation and its consequences”. Thus, by fusing *S3* and *S5* we can have different facets of the fact “Russian investigation”.

The above problem is best handled using semantic summarization [12, 19]. The idea is to extract facts, represented as triples, from text documents and model them as a graph. The nodes of the graph can be either words or word types. Each edge connecting two nodes represents their consecutive occurrence in the same sentence of the original text. Summary sentences are then generated using the top ranked paths in terms of grammaticality and fact coverage. The

graph model facilitates the retrieval and the fusion of important triples from both main and subordinate clauses. Additionally, facts are naturally connected along the paths with their facets. So, finding the best paths would automatically lead to finding the most important facets to be included in the summary. The main limitations are related to how this approach was applied to news summarization by Li et al. [12]. First, fact fusion merges triples having similar word types leading, in some cases, to incorrect results. For example, “Trump” and “Obama” both are of type person but they are two different entities and therefore sentences containing them should not be fused together. Second, long paths covering many triples are not necessarily the best since they might concatenate unrelated facts. Third, facts are clustered using predefined themes, which is inflexible for the dynamic nature of news content. In this paper, we aim at tackling the above problems extending the approach proposed by Li et al. [12] to handle news summarization in a more effective way. Our main contributions are as follows:

1. We propose a fact fusion strategy based on entity linking and predicate similarity. We perform entity linking via entity recognition, name normalization, and coreference resolution using Stanford NLP and DBpedia Spotlight, whereas predicate similarity is done using WordNet::Similarity.
2. In addition to the grammaticality and fact coverage, we employ node degrees to rank paths. This results in boosting paths having multiple authoritative nodes and therefore finding important facts to be included in the summary.
3. We propose an alternative to the predefined classification of facts by employing a dynamic grouping using K-means clustering. To this end, we use word2vec [17] trained on the Google News dataset to generate word vectors which are then used to cluster similar facts.
4. We run extensive experiments using the DUC’04 and DUC’07 datasets, showing that our approach outperforms the baseline approaches with a large margin in terms of ROUGE and PYRAMID scores.

2 Related Work

Our work falls into the category of abstractive summarization of news articles. Many previous attempts were based on facts extracted from main clauses [6, 10, 22, 26], in contrast to ours, that aims to enrich summaries with facets obtained from subclauses. Moreover, they did not focus on obtaining new facts by fusing together individual facts, instead they simply merged or clustered similar facts. Nevertheless, some core components of these approaches are related to our work. The first one is a fact extraction technique, which is either done via Open Information Extraction (OIE) such as OLLIE [25] and ClausIE [7], Semantic Role Labeling such as SEMAFOR [11] and SENNA SRL [5], or constituent/grammatical dependency parsing. The second one is a fact merging or clustering technique, which can be adopted for a triple fusion approach. Vanderwende et al. [26] used dependency parsing to extract simple facts, leveraged triple similarity, entity coreference and event coreference for clustering, and then generated summary sentences by unfolding the entity fragments and event

fragments. Similarly, D’Aciarno et al. [6] and later Amato et al. [1] leveraged dependency parsing for fact extraction and employed similarity measures based on the subject, predicate, and object for fact merging. Pighin et al. [22] developed their own data-driven pattern extractor for their basic semantic unit, and was able to merge multiple facts expressed in different ways.

Khan et al. [10] developed an approach that exploits SENNA [5] to extract basic semantic unit. They used WordNet::Similarity [21] for SRL unit clustering and then selected their representative fact from each cluster via a scoring function obtained by a genetic algorithm. Summary sentences were generated using SimpleNLG [8] with several heuristics employed. Genest and Lapalme [9] used dependency parsing for fact extraction in the form of INIT (Information Unit), another triple-like structure. Then, they produced summary sentences using a text-to-text generation based on SimpleNLG. Khan et al.’s BSU and Genest & Lapalme’s INIT can be annotated with dative or locative information, which makes their representation richer than the other approaches described above. Similarly, Li [13] used a simple BSU parsing to extract simple facts, which later are clustered based on semantic relatedness between concepts, similarity between verbs, and sentence co-occurrence. The sentence generation is a series of subject, predicate, and object unfoldings from the BSUs. Unfolded objects can be in the form of subclauses. However, fine-grained facts from the subclauses are not specifically extracted and merged.

On the other hand, we found fewer attempts that exploit fact extraction from subclauses. Bing et al. [3] used constituent parsing to extract noun phrases (NPs) and verb phrases (VPs), which then, being separately clustered, are combined after checking whether some NPs and VPs together satisfy constraints such as compatibility and validity. As a result, some NPs can be matched to VPs obtained from different sentences, forming new, potentially more informative summary sentences. Additionally, NPs (VPs) appearing in a constituent subtree up to two levels on a path containing only NP (VP) nodes are also taken into account, effectively parsing independent subclauses, but not dependent ones. Moreover, the approach did not address grammaticality when forming new sentences, and the new sentence formation is limited to coreference resolution of entities. The most relevant work to ours is by Li et al. [12], who developed a pattern-based approach to extract abstractive summaries from news articles. In contrast to Bing et al.’s, Li et al.’s approach is orthogonal to the type of subclause (dependent or independent). It also employs a grammaticality score in order to maintain the quality of its summary sentences. In the next section, we will explain in more detail Li et al.’s work and how our work is built upon it.

3 Background

Our work is based on the approach developed by Li et al. [12]. The main idea is to identify patterns of triples and further build summaries based on sentences having the largest number of patterns. Practically, triples are extracted from news articles, using OLLIE [25], in the form of $\langle \text{arg1}; \text{predicate}; \text{arg2} \rangle$, for example $\langle \text{Donald Trump}; \text{fired}; \text{FBI director} \rangle$. Then, a pattern is generated from each

triple, annotating the heads of arguments `arg1` and `arg2` with their types. The annotation is done using Stanford NER [15] and SEMAFOR [11]. For example, annotating the triple `<Donald Trump; fired; FBI director>` generates the pattern `<PROTAGONIST; fired; PERSON>`. As OLLIE may return clausal arguments, patterns with such arguments do not get their head annotated. For example `<PERSON; killed by; a gunman who is on the loose>` has only one argument annotated but it is a valid pattern. Further, patterns are clustered into predefined themes specific for TAC 2010 guided summarization task. Examples of such themes include, “what happened”, “reason”, “damages”, and “counter-measures”. Then, from each cluster, a graph is constructed by fusing together all patterns in the cluster based on their POS and lemma, ignoring stop words. Finally, sentences are generated by traversing all possible paths in the graph, and then ranked based on their grammaticality and pattern coverage. The top ranked path in each cluster is then picked as the representative summary sentence for the cluster.

We observe that the approach by Li et al. [12] has three main problems. The first one is that it uses a predefined list of “themes” to group patterns, which is specific for some, but not all kinds of news articles. Therefore, we need to develop an alternative way to cluster patterns when the themes are unknown. Second, they rely on POS and lemma for pattern fusion. We think that this is problematic because it can potentially fuse unrelated patterns, which consequently generate incorrect sentences. For example, consider the two patterns: `<Trump; told; Comey to stop his investigation about Russia>` and `<Trump; fired; Comey because of his investigation about Hillary Clinton>`. The graph fusion approach will generate the following result:

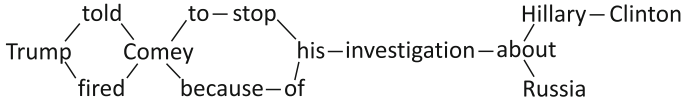


Fig. 1. An example of a fusion graph

Because the sentence ranking is based on pattern coverage and grammaticality, the chosen sentence would be “Trump told Comey to stop his investigation about Hillary Clinton” which is not correct w.r.t. the original news articles. The third problem is that pattern fusion relies on object typings returned by Stanford NER and SEMAFOR. Consider if Trump and Obama both appear in the original text. Both are of type PERSON, so during the fusion they will be merged together, which is not something that should happen because it often leads to incorrect summary sentence generation.

4 Improvements and Extensions

Our summarization approach follows the same line as the work of Li et al. [12], which we consider a baseline solution. Figure 2 shows an overview of our

approach. Triple extraction and grouping, graph fusion, and ranking modules indicate the steps that are improved from the baseline. On the other hand, entity and verb linking modules indicate our extensions of the baseline.

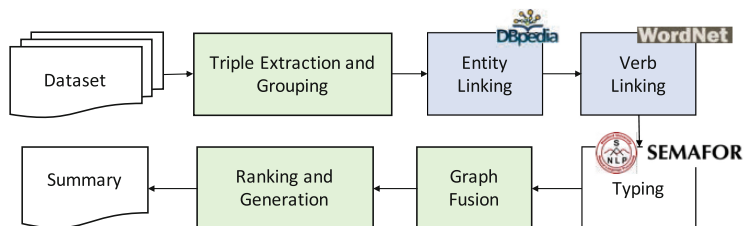


Fig. 2. Improvements and extensions

4.1 Triple Extraction and Grouping

We start by extracting triples from a set of news articles. Similarly to the baseline approach, we use OLLIE [25] to extract triples of the form $\langle \text{arg1}; \text{predicate}; \text{arg2} \rangle$. The summarization process starts by finding groups of similar triples, as it is crucial to find the sentences that have the same focus of the news. In other words, we aim at finding triples that tackle similar facets. Consider these triples:

T1: $\langle \text{Donald Trump}; \text{fired}; \text{FBI director James Comey in order to stop the Russian investigation} \rangle$

T2: $\langle \text{Democrats}; \text{said}; \text{FBI director James Comey was fired upon his handling of the investigation into Hillary Clinton's email server} \rangle$

T3: $\langle \text{Donald Trump}; \text{fired}; \text{FBI director James Comey the White House said in a statement Tuesday evening} \rangle$

We observe that $T1$ and $T3$ have the same first argument, the same predicate, and the same head of the second argument. Thus, they can be considered as most similar triples. However, $T1$ and $T2$ are more similar because both talk about the reason of the dismissal of Comey. The baseline approach tackled this problem by first grouping triples into predefined themes, such as “consequences” and “reasons”, relying on training data. Since this solution does not cover all news article datasets and is not flexible considering the dynamic nature of news, we propose an unsupervised approach to define triple groups (i.e., themes).

Our approach consists of three main steps. First, we use word2vec [17] trained on the Google News dataset to generate word embeddings for each word in each triple. Second, we enhance the generated word vectors by using PCA (Principal Component Analysis) as it was shown by [2] that this weighting improves the effectiveness of textual similarity tasks by 10% to 30%, and outperforms sophisticated supervised methods. Third, we perform K-means clustering, a well

established technique in machine learning, to create K clusters of similar triples based on the generated word vectors. The clustering technique starts by selecting randomly K triples as centroids and then maps all triples to the most similar centroid. The centroids then get updated and the process repeats until we obtain stable clusters. The number K of clusters reflects the size of the summary in terms of number of sentences. So, one representative sentence, which is a set of triples, is selected from each cluster to be part of the final summary.

4.2 Entity Linking and Predicate Similarity

When finding similar triples, mentioned entities are important. The heads of triple arguments are typically entities that can be either a person, an organization, a location, or any well-defined concept. The first issue with entity recognition is that existing tools do not have an agreement on what an entity is and therefore they might miss some important entities such as “Crimea”. The second issue is that entities are not always mentioned using their full names, but sometimes using abbreviations or only last names of people, which we call aliases. Interestingly, traditional Named Entity Recognition (NER) tools are not always able to recognize entities from aliases. The third issue is that NER approaches are not designed to detect entities that appear as coreferences. This is a problem for our work since we need to find similar triples. For example, there is no way to detect that the triples $\langle \text{Donald Trump}; \text{fired}; \text{FBI director James Comey} \rangle$ and $\langle \text{He}; \text{fired}; \text{Comey} \rangle$ have identical meaning because the entities are not resolved.

To overcome the above problems we follow our approach in [23], where we performed entity linking. We start by doing entity recognition, where we exploit DBpedia Spotlight [16], a large-scale knowledge base extracted from Wikipedia. It is a graph database that uses the RDF format. It represents Wikipedia categories as resources and uses the *rdf:type* predicate to state whether a resource is a class or an individual of a class. Using this property, we filter entities by removing all results produced by NER tools that have no property *rdf:type* in DBpedia Spotlight. Then, we introduce a name normalization technique that converts all aliases to normalized names to facilitate entity extraction. To begin, we extract entities from the news article using the entity filtering technique described earlier. For entities of type Person, we set as aliases first names, middle names, and last names. For other types, we find possible aliases using DBpedia Spotlight. As last step, we apply the Stanford Deterministic Coreference Resolution System [24] to map coreferences to their corresponding entities.

Another problem related to triple similarity are predicates. Some predicates, which are typically represented as verbs, have the same meaning. For example, the two triples $\langle \text{Donald Trump}; \text{fired}; \text{FBI director James Comey} \rangle$ and $\langle \text{Donald Trump}; \text{dismissed}; \text{FBI director James Comey} \rangle$ are basically the same. However, this cannot be detected if the two predicates are considered as two different words. To solve this problem, we use WordNet::Similarity [21] to detect similar predicates and use only one representative word for them. Since WordNet returns a similarity score for each pair of predicates, we set the similarity threshold high, concretely 90%, so we only fuse verbs (predicates) that have very close meanings.

4.3 Fusion Graph and Strict Merging

As a first step, we follow the baseline approach to build a fusion graph for each group of similar triples or patterns. At the beginning we use patterns since we strictly follow the baseline approach. The graph is constructed by iteratively adding patterns to it, as shown in Algorithm 1. A node is added to the graph for each word (token) in the pattern, where consecutive words are linked with directed edges. When adding a new pattern, a token from the pattern is merged with an existing node in the graph providing that they have the same POS tag and they share the same lemma. An essential observation is that some words such as “he” and “his” have the same POS tag “PRP” and the same lemma, but they should not be merged together. Also, stopwords like “the”, “to”, and “of” should not be merged together in order to avoid noise. It is important to clarify that without annotation, the core of each pattern is simply a sentence of the original text. The structure of triples is used only to identify their predicates and the arguments, to perform head argument annotation and triple similarity checking.

Algorithm 1. Graph Fusion

Data: P : set of patterns
Result: (V, E) : graph of fused patterns

```

1 ConstructGraph( $P$ )
2 begin
3    $(V, E) := (\emptyset, \emptyset)$ ; // empty graph
4   foreach  $p \in P$  do
5      $prevV := null$ ;
6      $ptokens := p.splitToTokens()$ ;
7     foreach  $t \in ptokens$  do
8       if  $t \notin V$  or  $isStopword(t)$  then
9          $curV := new\ Vertex(t)$ ;
10         $V := V \cup \{curV\}$ 
11      else
12         $curV := G.getV(t)$ ;
13      if  $prevVertex \neq null$  then
14         $e := new\ Edge(prevV, curV)$ ;
15         $E := E \cup \{e\}$ ;

```

We enhance the fusion graph by merging triples without annotation taking into account entity linking and predicate similarity when adding nodes. In other words, a token can now be entities or predicates, and therefore their linkings or similarities are involved during the merging process (the $t \notin V$ in line 8–10 in Algorithm 1). We also employ strict merging, where merging is done only for matching entities and predicates, but not for other types of nodes. The idea is to avoid topic drift and concatenating triples that are not compatible. The example in Sect. 3 show that the fusion of the two triples $\langle \text{Trump}; \text{told}; \text{Comey to stop his investigation about Russia} \rangle$ and $\langle \text{Trump}; \text{fired}; \text{Comey because of}$

his investigation about Hillary Clinton) might lead to the sentence “Trump told Comey to stop his investigation about Hillary Clinton” which is not correct.

4.4 Summary Sentence Selection

Sentences that compose the final summary are selected from the fusion graph. One path corresponds to one sentence. Paths are ranked based on two criteria: their grammaticality and their triple (or pattern) coverage. So, highly ranked paths should cover many paths which means that they summarize several facets of the same fact. Moreover, they should be grammaticality correct.

We implemented our own grammatical checker based on Stanford NLP and languagetools.org,¹ since the model used by the baseline was not publicly available. We performed a partial grammatical fix, focusing on the dangling verbs, i.e. verbs that are not correctly anchored to a subject, that are results of either OLLIE or the graph fusion. The fix is done by transforming the verb phrase into a well-formed clause using a relative pronoun (which, that, who, where, etc.) or a participle by analyzing the grammatical dependency to detect the occurrence of the dangling verbs and entity typing to determine the correct pronoun. Additionally, we analyze whether a dangling verb should be a passive voice by checking whether there exists a preposition that is connected to the verb as a nominal modifier. Finally, sentences without verbs are discarded.

We further enhanced path ranking exploiting, in addition to pattern coverage and grammaticality, node degrees. A node degree is the total number of both incoming and outgoing edges of the node. The idea is to select a path that has multiple important nodes, which are nodes having high degrees. Practically, our path ranking algorithm is a multi-step pairwise comparison in the following order: (1) pattern coverage, (2) node degree, and (3) grammaticality. For the node degree step, we compare first the average degree then the total degree of two paths. Finally, leveraging our grammaticality checker and fixer model, we set higher precedence in the following order: originally grammatical paths, grammatically fixable paths, and ungrammatical, non-fixable path.

5 Experiments

5.1 Setup

Datasets. For our evaluation, we used the DUC’04² and DUC’07³ datasets, which represent one of the most important English corpora for summarization. The DUC’04 contains 50 news topics while the DUC’07 dataset provides 45 news topics. Each news topic contains 10 news articles and 4 human summaries. We also prepared a dataset for manual assessment of the quality of our

¹ <https://www.languagetool.org/>.

² <http://duc.nist.gov/duc2004/tasks.html>.

³ <http://duc.nist.gov/duc2007/tasks.html>.

summaries. The code of our work can be found in <https://gitlab.inf.unibz.it/rprasojo/summarization>.

Assessment. The results are assessed automatically. We basically compare the summaries generated by the different approaches under comparison with the human summaries. In the case of manual assessment done on the randomly generated 100 summary sentences, we proceeded as follows. We asked 20 students and researchers in our faculty to independently assess the coherence and correctness of the summary sentences on a scale between 1 to 5. After that, we computed the average score of each sentence. The correctness of sentences regards whether the reported information corresponds to what really happened. By contrast, coherence is about the correctness of the sentence structure.

Strategies Under Comparison. We used the approach by Li et al. [12] as the baseline for our experiments. This approach represents the starting point of our work. We performed further improvements and tested the impact of each extension on the results. So, we have the following strategies under comparison:

1. **B.** The baseline approach by Li et al. [12];
2. **B+ EL.** The baseline approach with Entity Linking;
3. **B+ PL.** The baseline approach with Predicate Linking;
4. **B+ EL+ PL.** The baseline approach with Entity and Predicate Linking;
5. **B+ EL+ PL-T.** The baseline approach with Entity and Predicate Linking but without Typing Annotation;
6. **B+ EL+ PL-T+SM.** The baseline approach with Entity and Predicate Linking, without Typing Annotation, and with Strict Merging;
7. **B+ EL+ PL+SM.** The baseline approach with Entity and Predicate Linking and Strict Merging.

Metrics. We have used the following measures in our evaluation:

1. **ROUGE.** The ROUGE measure [14] consists in computing the overlap between automatically produced summaries and human produced summaries, which are considered as ground truth. The overlap between summaries is typically in terms of n -grams, where n is defined by the experiment setting. In our work, we used n -grams of size 1 and 2. The ROUGE metric is represented by two quantitative measures: *Recall* and *Precision*. We compute the *Recall* as the number of overlapping n -grams divided by the total number of n -grams present in the human produced summary. By contrast, the precision is given as the number of overlapping n -grams divided by the total number of n -grams in the automatic summary. In our experiments, we compute the *F1* measure, that combines both precision and recall.
2. **PYRAMID.** PYRAMID scoring [18] involves semantic matching of Summary Content Units (SCUs), so it can recognize semantically synonymous facts. We use the automated version proposed in [20], which leverages a weighted factorization model to transform the n -grams within sentence bounds of the generated summary, and the contributors and label of an SCU

into 100 dimensional vector representation. If the similarity between an n -gram vector of a summary and an SCU exceeds a given threshold, then the SCU is assigned to the summary. We use the same setting described in [3], including the two threshold values 0.6 and 0.65.

5.2 Results

The overall results of our approach are shown in Tables 1 and 2. The ROUGE scores are shown in Table 1 for all strategies and datasets. We observe that our approach improves significantly the precision and $F1$ measure over the baseline approach. For the DUC'04 dataset we have an increase of precision of 7% and of $F1$ measure of 11% considering unigram matching R-1. These values naturally decrease for bigram matching R-2, but we still improve the precision and $F1$ measure by 2% and 3% respectively. The same observations hold for DUC'07 with very similar values for unigram matching R-1. We notice that the improvement is a bit higher for bigram matching R-2, where we have an increase of precision and $F1$ measure that is no less than 3%. Having a closer look at the results of the different strategies we implemented, we observe that all our extensions improves precision and $F1$ measure with respect to the baseline approach. Each added extension increases the precision and $F1$ measure with a minimum of 1%. We just note a very slight decrease in $F1$ measure for both datasets depending on whether we use typing annotations or not.

Table 1. ROUGE scores in %, where B = Baseline, EL = Entity Linking, PL = Predicate Linking, T = Typing, and SM = Strict Merging

Method	DUC'04				DUC'07			
	R-1		R-2		R-1		R-2	
	R	$F1$	R	$F1$	R	$F1$	R	$F1$
B	32.65	27.21	8.31	7.12	30.81	26.12	7.01	6.89
B+EL	36.73	36.14	10.11	9.64	34.61	34.23	9.19	8.83
B+PL	34.44	33.89	9.66	8.43	31.55	33.07	8.38	8.12
B+EL+PL	37.36	36.82	10.64	9.77	35.97	36.00	9.63	9.44
B+EL+PL-T	38.21	37.33	10.68	9.89	36.88	36.61	9.70	9.61
B+EL+PL-T+SM	38.76	38.81	10.73	10.01	37.52	37.97	9.77	9.89
B+EL+PL+SM	39.15	38.77	10.95	9.95	37.91	37.62	10.02	9.80

We further computed the PYRAMID scores for the DUC'07 dataset as shown in Table 2. This dataset was the only one providing a ground truth with semantic annotation for PYRAMID scoring, while DUC'04 does not provide such a ground truth so a PYRAMID evaluation for it is not possible. We observe that our approach provides a significant improvement over the baseline that goes up to 25%.

Besides the ROUGE and PYRAMID score, our evaluation shows how the contentedness of the graph evolves as a result of our improvements. Initially, the

Table 2. PYRAMID (DUC'07 in %)

Method	<i>T:0.6</i>	<i>T:0.65</i>
B	48.12	40.90
B+EL	65.47	56.89
B+VL	61.71	53.75
B+EL+VL	69.94	61.08
B+EL+VL-T	71.22	62.65
B+EL+VL-T+SM	72.66	63.41
B+EL+VL+SM	73.04	63.88

baseline relies on the typing in the graph fusion, which causes a highly liberal merging. For instance, if a cluster of patterns contain many <PERSON>, even though not referring to the same person, then some of the paths will be long (i.e. high pattern coverage), with many of them potentially resulting in an incorrect merging. By leveraging EL+VL, the typings are replaced with the corresponding entity and verb annotations. This causes less liberal merging. Evidently, after applying EL+VL, the average pattern coverage score goes down from 6.32 to 3.29 with standard deviation down from 3.02 to 0.87, so our graph becomes more compact and less convoluted. Combined with the ROUGE and PYRAMID scores, we can be confident that most paths that are “fixed” from the baseline are bad paths. Less variance on the pattern coverage score also means more usage of node degree ranking tiebreakers. We measured that it rises from 11% (baseline) to 80% (B+EL+VL+SM).

We also give an example of two summaries, one generated by the baseline approach and one by our approach. The summaries are of the news articles talking about Donald Trump firing the FBI director.

Baseline. Donald Trump to leave office over the investigations into his administration’s links with Russia, a former National Security Agency said. Comey informing Congress that the FBI didn’t find anything and continued to believe Clinton’s practices did not merit the pursuance of any criminal charges. Comey was fired now at a time before he made Tuesday’s decision. He told the Senate last week it had made him “mildly nauseous” to think his intervention could have affected the election. Senator Richard Burr stopped conducting its own investigation into Russian meddling in the 2016 election. Other congressional committees are investigating a possible Russian connection mostly behind closed doors. Trump’s decision means that the bureau is conducting a search for a new director which will begin immediately. White House press secretary Sean Spicer learned of his dismissal from televisions said law enforcement veteran who had been critical of the Justice Department under former President George W. Bush to the top domestic investigative and surveillance organization.

B+EL+VL+SM. The US President Donald Trump was to face an indictment over allegations his campaign team colluded with Russia to disrupt the presidential election, which could put an end to his presidency. Today President Donald J. Trump informed FBI Director James Comey that he has been terminated and removed from office said in a statement Tuesday evening. The White House said that the impetus for the firing of Comey came from Rosenstein, who accused Comey of attempting to “usurp the attorney general’s authority” by publicly announcing why he felt the case should be closed without prosecution. Comey learned of his dismissal from television law enforcement sources. Mrs Clinton lays part of the blame for her shock election defeat last November on Mr Comey. Mr. Comey’s bungling of the investigation into Hillary Clinton’s private email server violated longstanding Justice Department policy. He told the Senate last week it had made him “mildly nauseous” to think his intervention could have affected the election. The House of Representatives and Senate intelligence committees are looking into the same allegations.

We observe that the summary provided by our approach has more correct sentences than the one by the baseline. More importantly, we can see a logical flow with our approach. The summary starts with the risks related to the Russian investigation, then moves smoothly to the consequence which is the dismissal of the FBI director. Then, it talks about the claimed motivation which is Clinton investigation, how Comey got the notification and how he felt. By contrast, the baseline summary talks about most of these issues but in almost random order.

5.3 Manual Evaluation

We manually assessed the coherence and correctness of our summary. These two metrics are best illustrated in the examples shown above. We can see that the sentence from the baseline “Comey was fired now at a time before he made Tuesday’s decision” is incorrect. Also the sentence “Donald Trump to leave office over the investigations into his administration’s links with Russian.” is incoherent. We ran our manual assessment on randomly selected 200 summary sentences, 100 for each approach (B and B+EL+VL+SM). Table 3 shows that our approach has a slightly better coherence and highly better correctness than the baseline.

Table 3. Manual evaluation result

Method	Coherence	Correctness
Baseline	4.08	2.61
B+EL+VL+SM	4.33	4.69

The assessors have a high degree of agreement, with an average standard deviation of 0.87 per sentence. There are very few instances of polar differences,

totaling 6 out of 200 sentences. The high coherence score for both approach shows that the graph fusion is able to keep coherence during the pattern merging process, even for the baseline. The assessors seem to give penalty to the coherence score when the sentence is less grammatical, which suggests that the partial grammatical fixer in our ranking model has some impact in increasing the coherence of the improved approach. On the other hand, the high increase in correctness suggests that the graph fusion is much more effective in correctly merging facts when leveraging entity and predicate linkings rather than typing.

6 Conclusions and Lessons Learned

We have proposed in this paper a summarization technique based on semantic triples and graph models starting from an existing baseline approach. We have proposed a series of improvements that help finding important facts mentioned in news articles together with their facets. We have shown that our linking techniques increase both the recall and F-measure. This suggests that our entity linking and predicate linking are more effective than the typing annotation used by the baseline in the graph fusion. Most of the entities and predicates were originally annotated with their types, causing incorrect merging during the fusion step. Our entity and predicate linking “replace” this annotation, which helps fixing incorrect merging. Removing the typing entirely seems to further increase the recall. However, adding strict merging on top of the typing annotation produces the best recall at the expense of the precision (and *F1* measure). This suggests that if entity and predicate linking are employed, merging based on typing annotation is still better in terms of recall than merging non-annotated tokens. In terms of PYRAMID scores, our approach produces more Summary Content Units than the baseline. This strengthens the ROUGE results, showing that our improvements help producing summaries with more informative content. We can also see from the manual assessment that our approach has high coherence and correctness scores.

Acknowledgment. This work has been partially supported by the project TaDaQua, funded by the Free University of Bozen-Bolzano.

References

1. Amato, F., d’Acierno, A., Colace, F., Moscato, V., Penta, A., Picariello, A.: Semantic summarization of news from heterogeneous sources. *Advances on P2P, Parallel, Grid, Cloud and Internet Computing. LNDECT*, vol. 1, pp. 305–314. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-49109-7_29
2. Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings (2016)
3. Bing, L., Li, P., Liao, Y., Lam, W., Guo, W., Passonneau, R.: Abstractive multi-document summarization via phrase selection and merging. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 1, pp. 1587–1597 (2015). (Volume 1: Long Papers)

4. Christensen, J., Soderland, S., Bansal, G., Mausam: Hierarchical summarization: scaling up multi-document summarization. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 902–912 (2014)
5. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **12**(Aug), 2493–2537 (2011)
6. d’Acerno, A., Moscato, V., Persia, F., Picariello, A., Penta, A.: Semantic summarization of web documents. In: 2010 IEEE Fourth International Conference on Semantic Computing (ICSC), pp. 430–435. IEEE (2010)
7. Del Corro, L., Gemulla, R.: ClausIE: Clause-based open information extraction. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 355–366. ACM (2013)
8. Gatt, A., Reiter, E.: SimpleNLG: A realisation engine for practical applications. In: Proceedings of the 12th European Workshop on Natural Language Generation, pp. 90–93. Association for Computational Linguistics (2009)
9. Genest, P.-E., Lapalme, G.: Framework for abstractive summarization using text-to-text generation. In: Proceedings of the Workshop on Monolingual Text-To-Text Generation, pp. 64–73. Association for Computational Linguistics (2011)
10. Khan, A., Salim, N., Kumar, Y.J.: A framework for multi-document abstractive summarization based on semantic role labelling. *Appl. Soft Comput.* **30**, 737–747 (2015)
11. Kshirsagar, M., Thomson, S., Schneider, N., Carbonell, J., Smith, N.A., Dyer, C.: Frame-semantic role labeling with heterogeneous annotations. In: ACL, vol. 2, pp. 218–224 (2015)
12. Li, P., Cai, W., Huang, H.: Weakly supervised natural language processing framework for abstractive multi-document summarization: weakly supervised abstractive multi-document summarization. In: Proceedings of the 24th CIKM, pp. 1401–1410. ACM (2015)
13. Li, W.: Abstractive multi-document summarization with semantic information extraction. In: EMNLP, pp. 1908–1913 (2015)
14. Lin, C.-Y., Hovy, E.: Automatic evaluation of summaries using n-gram co-occurrence statistics. In: Proceedings of the 2003 NAACL, vol. 1, pp. 71–78. Association for Computational Linguistics (2003)
15. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The Stanford coreNLP natural language processing toolkit. In: ACL (System Demonstrations), pp. 55–60 (2014)
16. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: DBpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems, pp. 1–8. ACM (2011)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
18. Nenkova, A., Passonneau, R.J.: Evaluating content selection in summarization: The Pyramid method. In: HLT-NAACL, vol. 4, pp. 145–152 (2004)
19. Oya, T., Mehdad, Y., Carenini, G., Ng, R.: A template-based abstractive meeting summarization: leveraging summary and source text relationships. In: Proceedings of the 8th International Natural Language Generation Conference (INLG), pp. 45–53. Association for Computational Linguistics, Philadelphia, June 2014
20. Passonneau, R.J., Chen, E., Guo, W., Perin, D.: Automated pyramid scoring of summaries using distributional semantics. In: ACL, vol. 2, pp. 143–147 (2013)

21. Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet::Similarity: Measuring the relatedness of concepts. In: *Demonstration Papers at HLT-NAACL 2004*, pp. 38–41. Association for Computational Linguistics (2004)
22. Pighin, D., Cornolti, M., Alfonseca, E., Filippova, K.: Modelling events through memory-based, open-IE patterns for abstractive summarization. In: *ACL*, vol. 1, pp. 892–901 (2014)
23. Prasojo, R.E., Kacimi, M., Nutt, W.: Entity and aspect extraction for organizing news comments. In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM 2015*, pp. 233–242. ACM, New York (2015)
24. Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., Manning, C.: A multi-pass sieve for coreference resolution. In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010*, pp. 492–501. Association for Computational Linguistics, Stroudsburg (2010)
25. Schmitz, M., Bart, R., Soderland, S., Etzioni, O., et al.: Open language learning for information extraction. In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 523–534. Association for Computational Linguistics (2012)
26. Vanderwende, L., Banko, M., Menezes, A.: Event-centric summary generation. In: *Working Notes of DUC*, pp. 127–132 (2004)
27. Wang, L., Raghavan, H., Castelli, V., Florian, R., Cardie, C.: A sentence compression based framework to query-focused multi-document summarization. arXiv preprint [arXiv:1606.07548](https://arxiv.org/abs/1606.07548) (2016)