Sentence Packaging in Text Generation from Semantic Graphs as a Community Detection Problem

Alexander Shvets
DTIC, UPF
alexander.shvets@upf.edu

Simon Mille
DTIC, UPF
simon.mille@upf.edu

Leo Wanner
ICREA and DTIC, UPF
leo.wanner@upf.edu

Abstract

An increasing amount of research tackles the challenge of text generation from abstract ontological or semantic structures, which are in their very nature potentially large connected graphs. These graphs must be "packaged" into sentence-wise subgraphs. We interpret the problem of sentence packaging as a community detection problem with post optimization. Experiments on the texts of the VerbNet/FrameNet structure annotated-Penn Treebank, which have been converted into graphs by a coreference merge using Stanford CoreNLP, show a high F_1-score of 0.738.

1 Introduction

An increasing amount of research in Natural Language Text Generation (NLTG) tackles the challenge of generation from abstract ontological (Bontcheva and Wilks, 2004; Sun and Mellish, 2006; Bouayad-Agha et al., 2012; Banik et al., 2013; Franconi et al., 2014; Colin et al., 2016) or semantic (Ratnaparkhi, 2000; Varges and Mellish, 2001; Corston-Oliver et al., 2002; Kan and McKeown, 2002; Bohnet et al., 2010; Flanigan et al., 2016) structures. Unlike input structures to surface generation, which are syntactic trees, ontological and genuine semantic representations are predominantly connected graphs or collections of elementary statements (as, e.g., RDF-triples or minimal predicate-argument structures) in which re-occurring elements are duplicated (but which can be, again, considered to be a connected graph). In both cases, the problem of the division of the graph into sentential subgraphs, which we will refer henceforth to as "sentence packaging", arises. In the traditional generation task distribution, sentence packaging is largely avoided. It is assumed that the text planning module creates a *text plan* from selected elementary statements (*elementary discourse units*), establishing discourse relations between them. The sentence planning module then either *aggregates* the elementary statements contained in the text plan into more complex statements or keeps them as separate simple statements, depending on the language, style, preferences of the targeted reader, etc. (Shaw, 1998; Dalianis, 1999; Stone et al., 2003). Even if data-driven, as, e.g., in (Bayyarapu, 2011), this strategy may suggest itself mainly for input representations with a limited number of elementary elements and simple sentential structures as target. In the context of scalable report (or any other narration) generation, which can be assumed to start, for instance, from large RDF-graphs (i.e., RDF-triples with cross-referenced elements), or from large semantic graphs, the aggregation challenge is incomparably more complex. In the light of this challenge and the fact that in a narration the discourse structure is, as a rule, defined over sentential structures rather than elementary statements, sentence packaging on semantic representations appears as an alternative that is worth to be explored. More recent data-driven concept-to-text approaches to NLTG, e.g., (Konstas and Lapata, 2012), text simplification, e.g., (Narayan et al., 2017), dialogue act realization, e.g., (Mairesse and Young, 2014; Wen et al., 2015), deal with sentence packaging, but, as a rule, all of them concern inputs of limited size, with at most 3 to 5 resulting sentence packages, while realistic large input semantic graphs may give rise to dozens. In what follows, we present a model for sentence packaging of large semantic graphs, which contain up to 75 sentences.

In general, the problem of sentence packaging consists in the optimal decomposition of a given

Proceedings of The 11th International Natural Language Generation Conference, pages 350–359,
Tilburg, The Netherlands, November 5-8, 2018.

graph into subgraphs, such that: (i) each subgraph is in itself a connected graph; (ii) the outgoing edges of the predicative vertices in a subgraph fulfil the valency conditions of these vertices (i.e., the obligatory arguments of a predicative vertice must be included in the subgraph); (iii) the appearance of a vertice in several subgraphs is subject to linguistic restrictions of co-reference.[1] In graph-theoretical terms, sentence packaging can be thus viewed as an approximation of *dense subgraph decomposition*, which is a very prominent area of research in graph theory. It has been also studied in the context of numerous applications, including biomedicine (e.g., for protein interaction network (Bader and Hogue, 2003) or brain connectivity analysis (Hagmann et al., 2008)), web mining (Sarıyuece et al., 2015), influence analysis (Ugander et al., 2012), community detection (Asim et al., 2017), etc. Our model is inspired by the work on community detection. The model has been validated in experiments on the VerbNet/FrameNet annotated version of the Penn TreeBank (Mille et al., 2017), in which coreferences in the individual texts of the corpus have been identified using the Stanford CoreNLP toolkit (Manning et al., 2014) and fused to obtain a graph representation. The experiments show that we achieve an F_1-score of 0.738 (with a precision of 0.792 and a recall of 0.73), which means that our model is able to cope with the problem of sentence packaging in NLTG.

The remainder of the paper is structured as follows. In Section 2, we introduce the semantic graphs that are assumed to be decomposed and analyze them. Section 3 outlines the experiments we carried out, and Section 4 discusses the outcome of these experiments. In Section 5, we briefly review the work that is related to ours. In Section 6, finally, we draw some conclusions and outline possible lines of future work.

2 Semantic Graphs

2.1 Overview

We assume a semantic graph to which the problem of sentence packaging is applied to be a labeled graph with *semantemes*, i.e., word sense disambiguated lexical items, as vertice labels and predicative argument relations as edge labels. The vertice labels are furthermore assumed to be typed in terms of semantic categories such as 'action', 'object', 'property', etc. A semantic graph of this kind can be a *Abstract Meaning Representation* (AMR) (Banarescu et al., 2013) obtained from the fusion of coreference vertices across individual sentential AMRs or a VerbNet or FrameNet structure obtained from the merge of sentential VerbNet respectively FrameNet structures that contain coreferences. An RDF-triple store which is annotated with semantic metadata, e.g., in OWL (https://www.w3.org/OWL/) can be equally converted into such a graph (Rodriguez-Garcia and Hoehndorf, 2018). Without loss of generality, we will assume, in what follows, that our semantic graphs are hybrid VerbNet / Framenet graphs in that we use first level VerbNet type ids / FrameNet type ids as vertice labels and VerbNet relations as edge labels.

As already mentioned in the Introduction, we use the VerbNet/FrameNet annotated version of the Penn TreeBank (henceforth *dataset*) to which we apply the co-reference resolution from Stanford OpenCore NLP to obtain a graph representation (and which we split into a development set and test set, with 85% and 15% texts that contained 78% and 22% of the sentences respectively). Consider the schematic representation of the semantic graph of one of the texts from the development set in Figure 1. It consists of two isolated subgraphs: one of them (to the left) comprises three sentences and the second (to the right) corresponds to a single sentence. The blue (dark) nodes correspond to verbal and nominal predicate tokens.

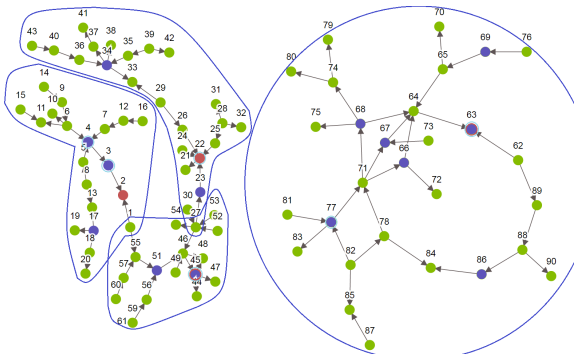

Figure 1: Example of a semantic graph of a text

As illustrated in Figure 2, a significant number (to be precise: 94%) of the text graphs obtained after the co-reference merge in the development set contain subgraphs which combine several sentences (in total, 77% of sentences were combined),

[1] Many more criteria apply, including language, style, topic, etc. In this work, we focus on formal criteria.

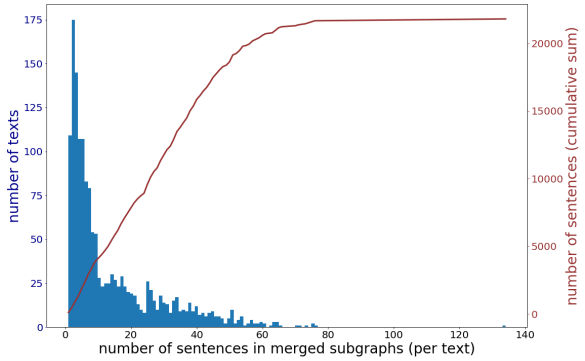

Figure 2: Sentence distribution in the graphs of the VerbNet/FrameNet annotated version of the Penn TreeBank

such that the task of sentence packaging is a necessary task in the context of NLTG. Even if the number of texts with a large number of merged sentences is relatively small, we can observe in Figure 2 that the line corresponding to the cumulative sum has a constant slope for the majority of texts, which implies that the number of sentences per bin of texts of the same size is close to constant. This means that each bin contributes evenly when we evaluate the quality of obtained packaging since we focus on recovering the sentences and assessing each of them individually, without averaging within a text.

2.2 Graph Analysis for Sentence Packaging

The generation information that characterizes a graph in the context of sentence packaging concerns: (i) the optimal number of sentences into which this given graph can be divided, and (ii) the profile (in semantic or graph theory terms) of a typical sentence of this graph. We use this information in the subsequent stages of sentence packaging.

2.2.1 Estimation of the Number of Sentences

In order to estimate the number of sentences into which a given semantic graph is to be decomposed, we built up a linear regression model with Ridge regularization on the development set with the features listed in the first column of Table 1. The statistics on chosen features are shown in the other columns, where Q_2 is a median, N_1 is an absolute number of sentences with a non-zero value of a parameter, and N_2 is a corresponding relative number.

The highest R^2-value was reached with the combination of all features, including FrameNet

	min	Q_2	**mean**	**max**	N_1	N_2
# tokens	2	17	17.5	95	28253	1.0
# edges	1	21	21.7	130	28253	1.0
# predicate nodes	0	11	11.3	67	28189	0.99
# argument nodes	1	12	12.6	67	28253	1.0
# roots	1	4	5	37	28253	1.0
# VerbNet nodes	0	3	3.2	15	26355	0.93
# Argument1	0	6	5.9	40	27794	0.98
# Argument2	0	4	4.3	30	27024	0.96
# Elaboration	0	2	2.2	19	22247	0.79
# NonCore	0	0	0.7	8	13415	0.47
# Set	0	0	1.2	26	12680	0.45

Table 1: Statistics of the features in the development set used for building up the linear regression model

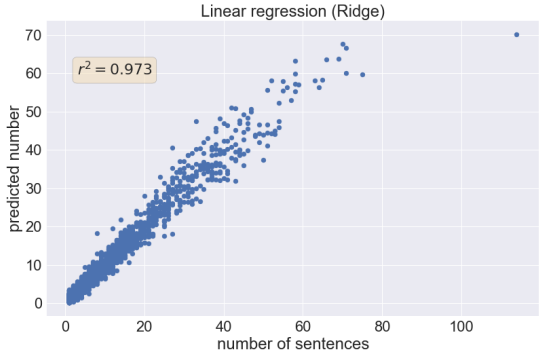

Figure 3: Predicting the number of sentences on the development set

and VerbNet classes of roots, which made R^2-value increase by 0.5 percentage point from 0.968 to 0.973; cf. Figure 3. The value is high, which means that the obtained model allows an accurate prediction of the number of sentences and can be used as an input parameter in community detection algorithms. We did not opt for using the number of predicates corresponding to different types for the regression since most of the types cover less than 7% of sentences from the development set.

2.2.2 Sentence Profiling

In order to obtain the prototypical profiles of the sentences in our dataset, we enriched the types of features used for the linear regression model above by features that play an important role in sentence formation: the type(s) of the parent node(s) of each node in the development set and the types of its arguments. With these enriched features at hand, we first built a multivariate normal distribution (MVN) of the most common non-correlated features of sentences chosen iteratively by cross-validation in such a way that a matrix of feature vectors is not singular for any set of folds. We

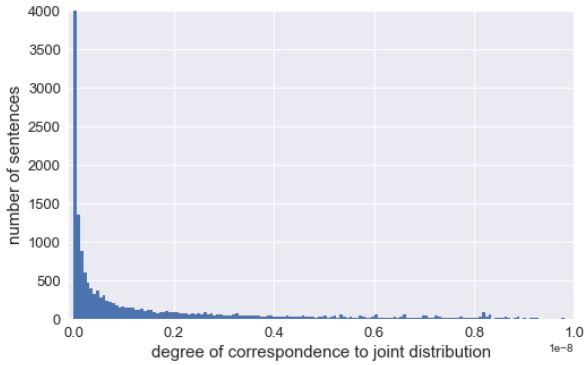

Figure 4: Correspondence of the sentences to the MVN distribution

ended up with the MVN distribution of 20 non-correlated features chosen from the top 100 features that appeared most frequently in sentences of 400 randomly chosen texts from the development set.

As an alternative to an iterative selection of the appropriate features, we applied Principal Component Analysis (PCA) (Jolliffe, 1986) to a space of the most common 100 features and selected principal vectors that describe 90% of the variance for building an MVN distribution. This step made the matrix of values of sentence features to be invertible, as required for the MVN distribution.

We assessed the proximity of the sentences of the development set to these initially obtained MVN distributions. As illustrated in Figure 4, the distribution of the degrees of correspondence to the joint distribution of 20 non-correlated features is right-skewed, with many sentences on the left that fit the distribution poorly. In order to remedy this, we implemented, for cases of a weak correspondence of a significant part of sentences (more than 15%) of the development set with the joint distribution, a clustering algorithm in a space of selected features (K-means, k=10) and built the distribution separately for each cluster. The proximity of the profile of the sentence being packaged has been assessed with respect to the joint distribution of each of the clusters – with success, as the results in Table 2, Subsection 3.2 below show.

3 Experiments

3.1 Background

Community detection aims to cluster a given social network (graph) into groups of tightly connected or similar vertices (Asim et al., 2017). The different algorithms which have been proposed are often adapted to the particular characteristics of the investigated network (Fortunato and Hric, 2016). Some algorithms take into account only the network structure (the mutual arrangement of vertices and the relationships between them) and are aimed at maximizing the modularity value (Blondel et al., 2008). Other algorithms consist in clustering the vertices by combining the most similar elements in terms of their attribute values without link analysis (Combe et al., 2015). Recently, the tendency has been to use both relationships between vertices and their characteristics and identify overlapping groups for optimal network decomposition (Yang et al., 2013). In our work, we experimented so far with algorithms which operate with links between vertices and allow for fast partitioning of huge graphs.

3.2 Setup of the Experiments

We first began to experiment with three community detection algorithms: LOUVAIN (Blondel et al., 2008), METIS (Karypis and Kumar, 2000), and COPRA (Gregory, 2010). However, already the first simple tests showed that COPRA performed poorly on our data in that it decomposed each graph into a small set of isolated subgraphs that did not include all the vertices of the original graph (see the exact figures below). Therefore, we discarded COPRA from further experiments, while LOUVAIN and METIS were taken to serve as baselines. Since METIS requires as input the number of communities (= sentences) into which a given graph is to be decomposed, we use linear regression presented in Subsection 2.2.1 as preprocessing stage.

To improve the quality of the initial decomposition made using community detection algorithms (i.e., our baselines) we carried out a local descent search, adding neighbour vertices to each subgraph one by one and keeping them if the correspondence of the subgraph to the multivariate distribution increased. The optimization is performed as a post-processing stage as follows:

1. for each $s \in S$, with S: = set of sentence subgraphs obtained by LOUVAIN / METIS
   (a) determine the degree of correspondence to the joint distribution (in case of several subgraphs, choose the most appropriate one) that is to be optimized.
   (b) apply local descent search, adding nodes from $s' \in S$ (with $s' \neq s$) iteratively

each time when it leads to the increase of the optimized parameter (subgraphs can share common nodes, i.e., overlap)

2. stop local descent search when there is no node that improves s.

F_1-score was chosen as a measure for the comparison of the quality of decompositions obtained by different algorithms on the test set. It is calculated for each original sentence since we consider a sentence as a separate unit. Its value takes into account which part of the original sentence was covered by the obtained subgraph and how many nodes that did not belong to the original sentence were mistakenly appended. Each isolated subgraph corresponds to one unit only, although it can include several original sentences. For those original sentences that are not captured in the majority of their nodes in any individual subgraph, F_1-score is equal to 0. The macro-F_1, i.e. the average F_1-score over all sentences, is a final measure.

The results are displayed in Table 2. 'No decomposition' stands for the case when any graph in the test set is considered to be a sentence (it can be considered as an additional baseline); 'METIS_{LR}' for "METIS with linear regression as a preprocessing stage", 'DC_{K}' for "descent search with K-means", and '$\text{DC}_{\neg\text{K}}$' "for descent search without K-means".

	Recall	**Precision**	**F_1-score**
No decomposition	0.313	0.264	0.274
LOUVAIN	0.69	0.726	0.68
METIS_{LR}	0.693	**0.814**	0.727
LOUVAIN+DC_{K}	0.707	0.709	0.681
LOUVAIN+$\text{DC}_{\neg\text{K}}$	0.705	0.704	0.678
LOUVAIN+PCA+ DC_{K}	0.701	0.714	0.681
METIS_{LR}+DC_{K}	0.73	0.792	**0.738**
METIS_{LR}+$\text{DC}_{\neg\text{K}}$	**0.731**	0.788	0.736
METIS_{LR}+PCA+ DC_{K}	0.714	0.795	0.731

Table 2: Results of testing the obtained models

As already mentioned above, COPRA showed a very poor performance on our data. The exact numbers were: mean recall = 0.113, mean precision = 0.088, and mean F_1-score = 0.084). Therefore, we did not include them into Table 2 and did not combine COPRA with other techniques.

4 Discussion

4.1 Performance Assessment

We can observe that the local descent search with the chosen optimization function leads to an increase of the mean F_1-score in each case. The use of a larger number of features with PCA leads to slightly poorer results, but still shows an improvement in comparison to the baseline community detection (LOUVAIN, and METIS_{LR}). However, METIS_{LR} is somewhat better than our optimizations with respect to precision and METIS_{LR}+$\text{DC}_{\neg\text{K}}$ is the best (even if by only a very minor margin, compared to the best F_1-score reaching METIS_{LR}+DC_{K}).

The very low figures for 'No Decomposition', i.e., the interpretation of each single graph as a sentence, show us that the problem of sentence packaging (or, in other words, decomposition of textual semantic graphs into sentential subgraphs) is indeed a relevant problem in large scale semantics-to-text generation.

Carrying out the error analysis, we assessed several obtained subgraphs in detail and identified at least two causes of the low values of precision and recall. The first cause lies in a suboptimal performance of the coreference resolution related to the merge of co-referenced nodes. For example, for the entity 'Mr. Peladeau', which appeared in a given text ten times, the module generated a node labeled 'Peladeau' and ten nodes labeled 'Mr.', connecting the 'Peladeau' node to all ten 'Mr.' nodes. This decreased our precision. We fixed the erroneous graphs by combining non-root nodes that were connected to the same input and output nodes with the same types of arguments and recalculated the measures. Some sentences were significantly affected by this change. For instance, for the mentioned example, the precision increased from 0.35 to 0.44. However, the overall mean F_1-score increased only by 0.5% because this error affected a relatively small number of subgraphs.

Another cause for poor quality of some obtained subgraphs consisted in the creation of subgraphs that contained subgraphs of several ground truth sentences. This led to the low value of precision, even if the recall was relatively high. To account for this problem, we defined a procedure that allowed us to separate such compound graphs into a set of subgraphs. This procedure duplicates those nodes that have two or more non-overlapping in-

put paths from roots which include a node with a defined VerbNet class. Since the output paths of duplicated nodes and the input paths without a node from VerbNet should not be necessarily assigned to all the copies of a node, we remove these paths to avoid overloading each single subgraph with redundant information.

The application of the node duplication procedure to the graphs obtained by LOUVAIN+PCA+DC$_K$ led to an increase of the overall mean precision (taking into account only covered ground truth sentences) from 0.85 to 0.96 and to a decrease of the recall from 0.86 to 0.67 since the procedure also affected some optimal sentence subgraphs by splitting them further into single clause subgraphs. At the same time, the coverage of the original sentences was improved (857 instead of 687 out of 908 were covered), which compensated the lower recall and led to an increase of the F_1-score by 10%. The potential values of precision and recall that could be reached if we combine subgraphs that belong to the same sentences are 0.91 and 0.77 respectively, which results in an F_1-score of 0.83. To tackle the problem of combining the subgraphs of clauses, full-text clustering could be used (Devyatkin et al., 2015). Adding back the removed paths linked to copied nodes will also contribute to the increase of overall quality of sentences.

4.2 Example

For illustration, consider in Figure 5 a subgraph obtained from a larger initial graph, which is shown in Figure 6 (the obtained subgraph is circled). The subgraph corresponds to the ground truth subgraph with a precision of 0.938 and a recall of 0.882. It might be seen that the obtained subgraph contains enough information to generate a sentence with a similar meaning as the original one.

The original sentence that corresponds to the subgraph in Figure 5 is *He said the company is experimenting with the technique on alfalfa, and plans to include cotton and corn, among other crops.*; cf. also Figure 7 for the text (with the corresponding sentence highlighted) captured by the initial graph. The text comprises 755 tokens of 41 sentences, which formed 10 isolated graphs after coreference resolution. The largest graph contains 578 vertices, which correspond to 32 sentences with 18 vertices that link sentences.

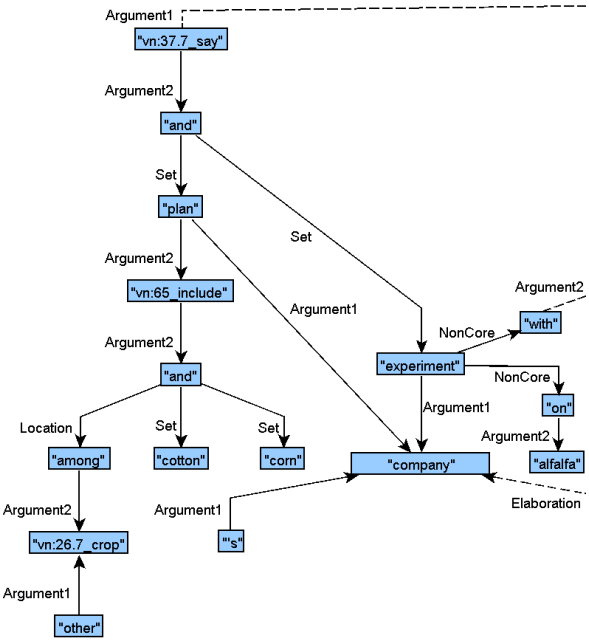

Figure 5: A sample subgraph extracted from a text graph

The LOUVAIN+PCA+DC$_K$ method applied to the whole graph detected 31 sentences out of 41. An additional separation of the obtained graphs by the procedure described above led to the detection of 9 extra sentences. Thus, the 98% of the ground truth sentences were recovered to a certain extent.

5 Related Work

A number of natural language text generators take as input sentence structures – for instance, sentence templates, as in the case of SimpleNLG generators (Gatt and Reiter, 2009), syntactic structures, as in the case of surface-oriented generators (Belz et al., 2011; Mille et al., 2018a), or more abstract semantic structures such as, e.g., AMRs; cf., e.g., (May and Priyadarshi, 2017; Song et al., 2018). For these generators, the problem of sentence packaging or aggregation is obviously obsolete. As already mentioned in the Introduction, in setups that start from input that is not yet cast into sentence structures, traditional NLTG foresees the task of (content) aggregation, which is dealt with as part of sentence planning (or *microplanning*): the elementary content elements, as assumed to be present in the text plan, are aggregated into more complex elements; see, among others, (Shaw, 1998; Dalianis, 1999; Stone et al., 2003; Gardent and Perez-Beltrachini, 2017).

Our work is more in line with Konstas and La-

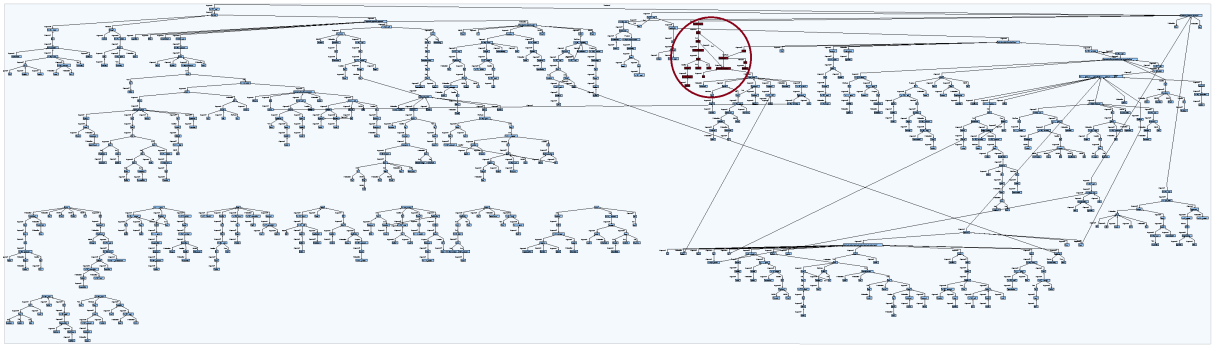

Figure 6: Example of the initial graph with one of the detected sentence subgraphs circled

116 - Researchers at Plant Genetic Systems N.V. in Belgium said they have developed a genetic engineering technique for creating hybrid plants for a number of key crops. 117 - The researchers said they have isolated a plant gene that prevents the production of pollen. 118 - The gene thus can prevent a plant from fertilizing itself. 119 - Such so - called male - sterile plants can then be fertilized by pollen from another strain of the plant , thereby producing hybrid seed. 120 - The new generation of plants will possess the flourishing , high - production trait known as `` hybrid vigor , '' similar to that now seen in hybrid corn. 121 - `` The development could have a dramatic effect on farm production , especially cotton , '' said Murray Robinson , president of Delta & Pine Land Co. , a Southwide Inc. subsidiary that is one of the largest cotton seed producers in the U.S.. 122 - On a commercial scale , the sterilization of the pollen - producing male part has only been achieved in corn and sorghum feed grains. 123 - That 's because the male part , the tassel , and the female , the ear , are some distance apart on the corn plant. 124 - In a labor - intensive process , the seed companies cut off the tassels of each plant , making it male sterile. 125 - They sow a row of male - fertile plants nearby , which then pollinate the male - sterile plants. 126 - The first hybrid corn seeds produced using this mechanical approach were introduced in the 1930s and they yielded as much as 20 % more corn than naturally pollinated plants. 127 - The vast majority of the U.S. corn crop now is grown from hybrid seeds produced by seed companies. 128 - A similar technique is almost impossible to apply to other crops , such as cotton , soybeans and rice. 129 - The male part , the anthers of the plant , and the female , the pistils , of the same plant are within a fraction of an inch or even attached to each other. 130 - The anthers in these plants are difficult to clip off. 131 - In China , a great number of workers are engaged in pulling out the male organs of rice plants using tweezers , and one - third of rice produced in that country is grown from hybrid seeds. 132 - At Plant Genetic Systems , researchers have isolated a pollen - inhibiting gene that can be inserted in a plant to confer male sterility. 133 - Jan Leemans , research director , said this gene was successfully introduced in oil - producing rapeseed plants , a major crop in Europe and Canada , using as a carrier a `` promoter gene '' developed by Robert Goldberg at the University of California in Los Angeles. 134 - The sterilizing gene is expressed just before the pollen is about to develop and it deactivates the anthers of every flower in the plant. 135 - Mr. Leemans said this genetic manipulation does n't hurt the growth of that plant. 136 - The researchers also pulled off a second genetic engineering trick in order to get male - sterile plants in large enough numbers to produce a commercial hybrid seed crop. 137 - They attached a second gene , for herbicide resistance , to the pollen - inhibiting gene. 138 - Both genes are then inserted into a few greenhouse plants , which are then pollinated and allowed to mature and produce seed. 139 - The laws of heredity dictate that half of the plants springing from these greenhouse - produced seeds will be male sterile and herbicide resistant and half will be male fertile and herbicide susceptible. 140 - The application of herbicide would kill off the male - fertile plants , leaving a large field of male - sterile plants that can be cross - pollinated to produce hybrid seed. 141 - Mr. Leemans said the hybrid rapeseeds created with this genetic engineering yield 15 % to 30 % more output than the commercial strains used currently. 142 - `` This technique is applicable to a wide variety of crops , '' he said , and added that some modifications may be necessary to accommodate the peculiarities of each type of crop. **143 - He said the company is experimenting with the technique on alfalfa , and plans to include cotton and corn , among other crops.** 144 - He said that even though virtually all corn seeds currently planted are hybrids , the genetic approach will obviate the need for mechanical emasculation of anthers , which costs U.S. seed producers about $ 70 million annually. 145 - In recent years , demand for hybrid seeds has spurred research at a number of chemical and biotechnology companies , including Monsanto Co. , Shell Oil Co. and Eli Lilly & Co. 146 - One technique developed by some of these companies involves a chemical spray supposed to kill only a plant 's pollen. 147 - But there have been problems with chemical sprays damaging plants ' female reproductive organs and concern for the toxicity of such chemical sprays to humans , animals and beneficial insects. 148 - However , Paul Johanson , Monsanto 's director of plant sciences , said the company 's chemical spray overcomes these problems and is `` gentle on the female organ. '' 149 - Biosource Genetics Corp. , Vacaville , Calif. , is developing a spray containing a gene that spreads from cell to cell and interferes with the genes that are responsible for producing pollen. 150 - This gene , called `` gametocide , '' is carried into the plant by a virus that remains active for a few days. 151 - Robert Erwin , president of Biosource , called Plant Genetic 's approach `` interesting '' and `` novel , '' and `` complementary rather than competitive. '' 152 - `` There is a large market out there hungry for hybrid seeds , '' he said. 153 - Mr. Robinson of Delta & Pine , the seed producer in Scott , Miss. , said Plant Genetic 's success in creating genetically engineered male steriles does n't automatically mean it would be simple to create hybrids in all crops. 154 - That 's because pollination , while easy in corn because the carrier is wind , is more complex and involves insects as carriers in crops such as cotton. 155 - `` It 's one thing to say you can sterilize , and another to then successfully pollinate the plant , '' he said. 156 - Nevertheless , he said , he is negotiating with Plant Genetic to acquire the technology to try breeding hybrid cotton.

Figure 7: Original plain text with the recovered sentence subgraph highlighted

pata (2012)'s data-driven concept-to-text model, which creates from the input database records hypergraphs that are then projected onto multiple sentence reports. We also depart from graphs (which we create from isolated semantic sentence structures by establishing coreference links between coinciding elements across different structures), only that we work with graphs that are considerably larger than those Konstas and Lapata work with (up to 75 resulting sentences per graph vs. >10 resulting sentences per graph). Furthermore, while we use community detection algorithms (and focus only on the problem of sentence packaging), they view the entire problem of the verbalization of a hypergraph as a graph traversal problem.

The difference in the size of the input data (and thus the number of the resulting sentences) is also a distinctive feature of our proposal when we compare it to other works that deal with sentence packaging. For instance, Narayan et al. (2017) split in their experiments on text simplification complex sentences into 2 to 3 more simple sentences. As content representation, they use the WebNLG dataset of RDF-triples (Gardent et al., 2017). To split a given set of RDF-triples into several subsets, they learn a probabilistic model. Wen et al. (2015) use LSTM-models to generate utterances from a given sequence of tokens in the context of a dialogue application.

Since for our experiments we apply coreference resolution to create from the VerbNet/Framenet annotated sentences of the Penn Treebank large connected graphs, our work could be also considered to be related to the recent efforts on the creation of datasets for NLTG; cf., e.g., (Gardent et al., 2017; Novikova et al., 2017; Mille et al., 2018b). However, so far, the coreference resolution has been entirely automatic, with no subsequent thorough validation and manual correction. Both would be needed to ensure high quality of the resulting dataset.

6 Conclusions and Future Work

We have presented a community detection-based strategy for packaging semantic (VerbNet/FrameNet) graphs into sentential subgraphs and tested it on a large dataset. We have shown that, in principle, sentence packaging can be interpreted as a community detection problem since community detection algorithms aim to identify densely connected subgraphs–which can be expected from sentential structures. The evaluation suggests that the subgraphs obtained by community detection can be further improved by a post-processing stage, e.g., by descent search or PCA.

The duplication of nodes for an additional decomposition of obtained graphs led to an increase of the performance. To avoid the unnecessary splitting of optimal subgraphs, as observed in some cases, the offered procedure might be furthermore restricted, for example, by duplicating only the nodes with high centrality measures.

In the future, we plan to explore community detection algorithms which will allow us to take the attributes of the vertices into account. For this purpose, the optimization function must be modified to take into account the mutual compatibility of vertices rather than their similarity, since vertices within one sentence usually have different properties and do not form homogeneous communities in a general sense. Furthermore, we plan to explore to what extent reinforcement learning-based graph partitioning algorithms that take the specifics of the semantic graphs into account in terms of features are suitable for the problem of sentence packaging.

Acknowledgments

The presented work was supported by the European Commission under the contract numbers H2020-645012-RIA, H2020-7000024-RIA, H2020-700475-IA, and H2020-779962-RIA and by the Russian Foundation for Basic Research under the contract number 18-37-00198. Many thanks to the three anonymous reviewers, whose insighful comments helped to improve the final version of the paper.

References

Yousra Asim, Abdul Majeed, Rubina Ghazal, Basit Raza, Wajeeha Naeem, and Ahmad Kamran Malik. 2017. Community detection in networks using node attributes and modularity. *Int J Adv Comput Sci Appl* 8(1):382–388.

G. Bader and C. Hogue. 2003. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics* 4(2).

L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the Linguistic Annotation Workshop*.

Eva Banik, Claire Gardent, and Eric Kow. 2013. The KBGen challenge. In *Proceedings of ENLG*. pages 94–97.

H.S. Bayyarapu. 2011. Efficient Algorithm for Context Sensitive Aggregation in Natural Language Generation. In *Proceedings of the Recent Advances in Natural Language Processing Conference*. pages 84–89.

Anja Belz, Michael White, Dominic Espinosa, Eric Kow, Deirdre Hogan, and Amanda Stent. 2011. The First Surface Realisation Shared Task: Overview and Evaluation Results. In *Proceedings of the 13th European Workshop on Natural Language Generation*. pages 217–226.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008(10):P10008.

Bernd Bohnet, Leo Wanner, Simon Mille, and Alicia Burga. 2010. Broad coverage multilingual deep sentence generation with a stochastic multi-level realizer. In *Proceedings of COLING*. Beijing, China, pages 98–106.

Kalina Bontcheva and Yorick Wilks. 2004. Automatic report generation from ontologies: the MIAKT approach. In *International conference on application of natural language to information systems*. Springer, pages 324–335.

Nadjet Bouayad-Agha, Gerard Casamayor, Simon Mille, and Leo Wanner. 2012. Perspective-oriented generation of football match summaries: Old tasks, new challenges. *ACM Trans. Speech Lang. Process.* 9(2):3:1–3:31. https://doi.org/10.1145/2287710.2287711.

Emilie Colin, Claire Gardent, Yassine Mrabet, Shashi Narayan, and Laura Perez-Beltrachini. 2016. The WebNLG challenge: Generating text from dbpedia data. In *Proceedings of INLG*. pages 163–167.

David Combe, Christine Largeron, Mathias Géry, and Előd Egyed-Zsigmond. 2015. I-louvain: An attributed graph clustering method. In *International Symposium on Intelligent Data Analysis*. Springer, pages 181–192.

Simon Corston-Oliver, Michael Gamon, Eric Ringger, and Robert Moore. 2002. An overview of Amalgam: A machine-learned generation module. In *Proceedings of INLG*. New-York, NY, USA, pages 33–40.

Hercules Dalianis. 1999. Aggregation in natural language generation. *Computational Intelligence* 15(4):384–414.

Dmitry Devyatkin, Ilya Tikhomirov, Alexander Shvets, Oleg Grigoriev, and Konstantin Popov. 2015. Full-text clustering methods for current research directions detection. In *DAMDID/RCDL*. pages 152–156.

Jeffrey Flanigan, Chris Dyer, Noah A Smith, and Jaime Carbonell. 2016. Generation from abstract meaning representation using tree transducers. In *Proceedings of NAACL:HLT*. pages 731–739.

Santo Fortunato and Darko Hric. 2016. Community detection in networks: A user guide. *Physics Reports* 659:1–44.

Enrico Franconi, Claire Gardent, Ximena Juarez-Castro, and Laura Perez-Beltrachini. 2014. Quelo natural language interface: Generating queries and answer descriptions. In *Natural Language Interfaces for Web of Data*.

Claire Gardent and Laura Perez-Beltrachini. 2017. A statistical, grammar-based approach to microplanning. *Computational Linguistics* 43(1):1–30.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating Training Corpora for NLG Micro-planning. In *Proceedings of ACL*.

Albert Gatt and Ehud Reiter. 2009. SimpleNLG: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation*. pages 90–93.

Steve Gregory. 2010. Finding overlapping communities in networks by label propagation. *New Journal of Physics* 12(10):103018.

P. Hagmann, L. Cammoun, X. Gigandet, R. Meuli, C. J. Honey, V. J. Wedeen, and O. Sporns. 2008. Mapping the structural core of human cerebral cortex. *PLoS Biology* 6(7):888–893.

Ian T Jolliffe. 1986. Principal component analysis and factor analysis. In *Principal component analysis*, Springer, pages 115–128.

Min-Yen Kan and Kathleen McKeown. 2002. Corpus-trained text generation for summarization. In *Proceedings of INLG*. New-York, NY, USA, pages 1–8.

George Karypis and Vipin Kumar. 2000. Multilevel k-way hypergraph partitioning. *VLSI design* 11(3):285–300.

I. Konstas and M. Lapata. 2012. Unsupervised concept-to-text generation with hypergraphs. In *Proceedings of NAACL*. pages 752–761.

F. Mairesse and S. Young. 2014. Stochastic Language Generation in Dialogue Using FLMs. *Computational Linguistics* 40(4):763–799.

C.D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S.J. Bethard, and D. McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. pages 55–60.

Jonathan May and Jay Priyadarshi. 2017. SemEval-2017 Task 9: Abstract Meaning Representation Parsing and Generation. In *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017)*.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, Emily Pitler, and Leo Wanner. 2018a. The First Multilingual Surface Realisation Shared Task (SR '18): Overview and Evaluation Results. In *Proceedings of the First Workshop on Multilingual Surface Realisation*. pages 1–12.

Simon Mille, Anja Belz, Bernd Bohnet, and Leo Wanner. 2018b. Underspecified Universal Dependency Structures, as Inputs for Multilingual Surface Realisation. In *Proceedings of the 11th International Conference on Natural Language Generation*.

Simon Mille, Roberto Carlini, Ivan Latorre, and Leo Wanner. 2017. UPF at EPE 2017: Transduction-based Deep Analysis. In *Shared Task on Extrinsic Parser Evaluation (EPE 2017)*. Pisa, Italy, pages 80–88.

Shashi Narayan, Claire Gardent, Shay B Cohen, and Anastasia Shimorina. 2017. Split and rephrase. *arXiv preprint arXiv:1707.06971* .

J. Novikova, O. Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*.

Adwait Ratnaparkhi. 2000. Trainable methods for surface Natural Language Generation. In *Proceedings of NAACL:HLT*. Seattle, WA, USA, pages 194–201.

M.A. Rodriguez-Garcia and R. Hoehndorf. 2018. Inferring ontology graph structures using OWL reasoning. *BMC Bioinformatics* 19(7).

A.E. Sarıyuece, C. Seshadhri, A. Pinar, and Ue.V. Çatalyuek. 2015. Finding the hierarchy of dense subgraphs using nucleus decompositions. *arXiv:1411.3312v2* .

James Shaw. 1998. Clause Aggregation Using Linguistic Knowledge. In *Proceedings of INLG*. pages 138–148.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A Graph-to-Sequence Model for AMR-to-Text Generation. In *Proceedings of ACL*.

Matthew Stone, Christine Doran, Bonnie L. Webber, Tonia Bleam, and Martha Palmer. 2003. Microplanning with Communicative Intentions: The SPUD System. *Computational Intelligence* 19:311–381.

Xiantang Sun and Chris Mellish. 2006. Domain independent sentence generation from rdf representations for the semantic web. In *Combined Workshop on Language-Enabled Educational Technology and Development and Evaluation of Robust Spoken Dialogue Systems*. Riva del Garda, Italy.

J. Ugander, L. Backstrom, C. Marlow, and J. Kleinberg. 2012. Structural diversity in social contagion. *Proceedings of the National Academy of Sciences* 109(16):5962–5966.

Sebastian Varges and Chris Mellish. 2001. Instance-based Natural Language Generation. In *Proceedings of NAACL*. Pittsburgh, PA, USA, pages 1–8.

Tsung-Hsien Wen, Milica Gašć, Nikola Mrkšć, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In *Proceedings of EMNLP*. pages 1711–1721.

Jaewon Yang, Julian McAuley, and Jure Leskovec. 2013. Community detection in networks with node attributes. In *Data Mining (ICDM), 2013 IEEE 13th international conference on*. IEEE, pages 1151–1156.