Unsupervised Semantic Abstractive Summarization

Shibhansh Dohare
CSE Department
IIT Kanpur
sdohare'

Vivek Gupta
Microsoft Research
Bangalore
t-vigu"

Harish Karnick
CSE Department
IIT Kanpur
hk'

Abstract

Automatic abstractive summary generation remains a significant open problem for natural language processing. In this work, we develop a novel pipeline for Semantic Abstractive Summarization (SAS). SAS, as introduced by Liu et al. (2015) first generates an AMR graph of an input story, through which it extracts a summary graph and finally, creates summary sentences from this summary graph. Compared to earlier approaches, we develop a more comprehensive method to generate the story AMR graph using state-of-the-art co-reference resolution and *Meta Nodes*. Which we then use in a novel unsupervised algorithm based on how humans summarize a piece of text to extract the summary sub-graph. Our algorithm outperforms the state of the art SAS method by 1.7% F1 score in node prediction.

1 Introduction

Summarization of large texts is still an open problem in natural language processing. Automatic summarization is often used in summarizing large texts like stories, journal papers, news articles and even larger texts like books and court judgments.

Existing methods for summarization can be broadly categorized into two categories *Extractive* and *Abstractive*. Most of the work done on summarization in the past has been Extractive Dang and Owczarzak (2008). Extractive methods directly pick up words and sentences from the text to generate a summary. Vanderwende et al. (2004) transformed the input to nodes, then used the Pagerank algorithm to score nodes, and finally grow the nodes from high-value to low-value using some heuristics. Some of the approaches combine this with sentence compression so that more sentences can be packed in the summary. McDonald (2007), Martins and Smith (2009), Almeida and Martins (2013), and Gillick and Favre (2009) among others used ILPs and approximations for encoding compression and extraction. However, human level summary generation require rephrasing sentences and combining information from different parts of the text. Thus, these methods are inherently limited in the sense that they can never generate human level summaries for large and complicated documents.

On the other hand, most Abstractive methods take advantages of the recent developments in deep learning. Specifically, the recent success of the sequence to sequence Sutskever et al. (2014) learning models, where recurrent networks read the text; encodes it and then generate target text produce promising results. Rush et al. (2015), Chopra et al. (2016), Nallapati et al. (2016), See et al. (2017) used standard encoder-decoder models along with their variants to generate summaries. Takase et al. (2016) incorporated the AMR information in the standard encoder-decoder models to improve results. These approaches have produced promising results and have been recently shown to be competitive with the extractive methods, but they are still far from reaching human level quality in summary generation. One of the significant problems with these methods is that there is no guarantee that they can handle subtleties of language like the presence of a word that negates the meaning of the full text, hard to capture co-references, etc.

Banarescu et al. (2013) introduced AMR as a base for work on statistical natural language understanding and generation. AMR tries to cap-

'@cse.iitk.ac.in, "@microsoft.com, Shibhansh is the corresponding author

Proceedings of ACL 2018, Student Research Workshop, pages 74–83
Melbourne, Australia, July 15 - 20, 2018. ©2018 Association for Computational Linguistics

ture *"who is doing what to whom"* in a sentence. An AMR represents the meaning of a sentence using rooted, acyclic, labeled, directed graphs. Figure 2 shows the AMR graph of the sentence *"I looked carefully all around me"* generated by the JAMR parser Flanigan et al. (2014). The nodes in the AMR are labeled with *concepts*, in Figure 2 *'around'* represents one such concept. Edges contain the information regarding the *semantic relation* between the concepts. In Figure 2 *direction* is the relation between the concepts *look-01* and *around*. AMR relies on Propbank for semantic relations (edge labels). Concepts can also be of the form *run-01* where the index *01* represents the first sense of the word *run*. Further details about the AMR can be found in the AMR guidelines Banarescu et al. (2015). Liu et al. (2015) started the work on summarization using AMR, which we call Semantic Abstractive Summarization (SAS).

Liu et al. (2015) introduced the fundamental idea behind SAS. In SAS the final summary is produced by extracting a summary subgraph from the story graph and generating the summary from this extracted graph (See Figure 1). But the work was limited to obtaining the summary graph due to the absence of AMR to text generators at that time. They used various graphical features like distance from the root, the number of outgoing edges, etc. and sentence number as features for nodes. The procedure then learned weights over these features with the constraint that the nodes must form a connected graph.

In this work, we propose an alternative method to use AMRs for abstractive summarization. Our approach is inspired by the way humans summarize any piece of text. User studies Chin et al. (2009); Kang et al. (2011) have shown that humans summarize by first writing down the key phrases and then try to figure out the relationships among them and then organize the data accordingly. Falke and Gurevych (2017) used similar ideas to propose the task of concept map based summarization. We design our algorithm along the same lines. The first step is to find the most important entities/events in the text. The second step is to identify the key relations among the most important entities/events, and finally, in the last step, we capture information around the selected relation. AMRs provide a natural way to achieve this process, as all the events/entities can be represented by a node Rao et al. (2017) or a group of nodes, while any relation can be captured by a path in the AMR graph. We also develop a more comprehensive method to generate the story AMR from the sentence AMRs based on event/entity co-reference resolution and *Meta Nodes*. Our algorithm outperforms the previous state of the art methods for SAS by 1.7% F1 score on Node prediction.

Our major contributions in this work are :

- We propose a novel unsupervised algorithm for the key step of summary graph extraction, which provides a stronger baseline for future work on SAS.
- We propose a novel method to generate the story AMR based on a more comprehensive co-reference resolution and *Meta Nodes*.

The rest of the paper is organized as follows. Section 2 and 3 contain description of the datasets and the algorithm used for summary generation respectively. Section 4 contains the results of experiments using our approach.

2 Datasets

We use the proxy report section of the AMR Bank Knight et al. (2014), as it is the only section that is relevant for the task because it contains the gold-standard (human-generated) AMR graphs for news articles and their summaries. In the training set, the stories and summaries contain 17.5 sentences and 1.5 sentences on average respectively. The training and test sets include 298 and 33 summary document pairs respectively.

3 Pipeline for Summary Generation

The pipeline consists of three major steps. The first step is to convert the document into an AMR (step-1). The next step is to extract a summary AMR from the document AMR constructed in the previous step (step-2). The final step generates text from the extracted sub-graph (step-3). In the following subsections, we expand on each step.

3.1 Step 1: Story to AMR: Document graph generation

Document AMR refers to the AMR representing the meaning of the whole document. The AMR

Code for the complete pipeline for the end to end summarization is available at https://github.com/shibhansh/Unsupervised-SAS

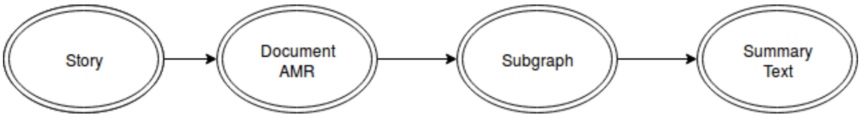


Figure 1: The pipeline proposed by (Liu et al., 2015) had the following step - AMR Parsing, Naive node merging, Subgraph selection and Text generation

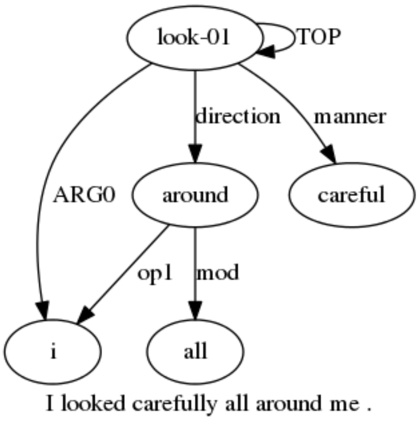


Figure 2: The graphical representation of the AMR graph of the sentence : *"I looked carefully all around me"* using AMRICA Saphra and Lopez (2015)

formalism guarantees that no two nodes refer to the same event/entity. Liu et al. (2015) extends this principle to multiple sentences by merging nodes referring to the same named entity (or date) across sentences. However, they adopted a naive approach to for co-reference resolution using a simple name and date matching 3.1. The co-reference resolution can be greatly improved if we take advantage of the huge literature on text co-reference resolution. We solve *node co-reference resolution* using *text co-reference resolution* followed by mapping the text to a node using *Alignments*.

Node Co-reference Resolution is a crucial step, as a wrongly generated document AMR can produce a factually wrong summary. To mitigate wrong mergers, we implement multiple sanity checks to avoid wrong mergers. Text co-reference resolution techniques can be broadly categorized into three major categories - neural, statistical and rule-based. We used the state-of-the-art end-to-end neural co-reference resolution system Lee et al. (2017). Future work can use an ensemble of co-reference resolvers to improve robustness. A list of major sanity checks that we employed

- Don't merge nodes which have an outgoing edge with label *:name* if the value of *:name* argument is different, and neither of the names is the initials of the other
- Don't merge if, cycle emerges in the graph after the merger
- Don't merge if, the nodes to be merged have common outgoing edge labels, and the nodes that are connected with these edges are different

For mapping text to the node, we use *alignments*. *Alignments* provide a mapping from a word in the text to the corresponding node in the AMR. Most co-reference systems provide co-references between noun phrases instead of individual words. But for *node co-reference resolution* we are required to merge individual nodes rather than a group of nodes. However, Lee et al. (2017) system also outputs attention weight for every word of a noun phrase which signifies the importance of each word in the noun phrase. We merge the nodes corresponding to the word that has the maximum attention weight among the words in the noun phrase.

Merging nodes that refer to the same event/entity suggests that the merged node is more important in the graph than the original nodes as there are more incoming and outgoing edges in the graph now. Co-reference resolution captures explicit reference of an event/entity, which implies that the nodes should be merged as they are same and thus it helps increase the importance of the node. But, there are many cases where words are not referring to the same entity or event but they refer to the same abstract concept, or there might be cases where the words are talking about the same event without explicitly referring to it. In such cases, these words should reinforce the importance of each other, but simple co-reference resolution does not capture this, and hence co-reference resolution is not enough. We need something new in the graph that captures when two nodes are reinforcing the importance of each other without actually merging the two nodes. In table 1 we present two examples where words reinforce the importance of each other without referring to the exact same thing.

These examples inspire us to introduce a new set of nodes which we call *Meta nodes*. In this work, we use *Meta nodes* to increase the importance of only common nouns. Common nouns like drugs, opium, etc. can occur a lot of times in the text which suggests that they are relevant for the text, but they are not identified by co-reference systems as their different occurrences do not refer to exactly the same thing. To capture the important *common nouns* which are otherwise not captured in the co-reference resolution, we add a new *Meta node* in the graph for each such set of *common nouns*. In Example 2 of Fig. 1, we introduce a *Meta Node* for the common noun *Opium*, which is present twice in the story. Each *Meta node* is connected to all the occurrences of the corresponding common noun. The nodes connected with a meta node signifies that the nodes at some level might refer to the same thing. *Meta nodes* are used as representative for the group during ranking but they are not extracted in the final summary graph, and hence they are not used during the final step of summary generation.

The cases that we examined in Table 1 are cases where the words don't have a perfect identity but rather a near identity. This points out that co-reference resolution is not a simple yes/no question but rather a complicated one. This problem of the complexity of co-reference resolution has been explored theoretically in the literature Recasens et al. (2011); Versley (2008) and our work will benefit directly from more work on the complexity of co-reference resolution. In our current work, we don't implement any procedure to detect reinforcements of the sort given in Example 2 of Table 1. Future works may include event co-reference resolution and word similarity using word embeddings to identify such reinforcements.

3.2 Step 2: Summary Graph Extraction

Summary graph extraction is a key step in SAS. In this step, we extract the summary sentence AMR graphs from the document AMR produced in Step-1. We take our cue from the way humans summarize a text by first identifying the most important entities/events in the text then finding the most important relationships among these events/entities and finally include information surrounding the selected relationship(s).

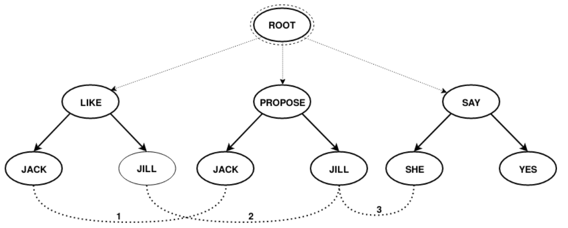


Figure 3: An example of node merging in a very basic AMR, The mergers *1* and *2* were also present in the methods proposed by Liu et al. (2015) but not the merger *3*. Here, dash line represent node to be merged.

Step-A: Finding Important Nodes - For finding important events/entities, we use *term frequency-inverse document frequency* $(\mathrm{Tf}-\mathrm{IDF})$ to determine the importance of any node. We first find the top n nodes in the graph using *term frequency*. This n depends upon the size of the summary required. Similar to earlier approaches we use *Alignments* to find text corresponding to the nodes. Finally, we use Tf-IDF values of the text corresponding to the nodes to rank the selected n nodes. The proxy report section of the AMR Bank is quite small with only 298 training stories. We use the CNN-Dailymail Hermann et al. (2015) corpus containing around 300,000 news articles to evaluate the Document Frequencies (DF). We calculate $\mathrm{Tf}-\mathrm{Idf}$ as -

$$\mathrm{Tf}-\mathrm{Idf}=\mathrm{Tf}\times\log_{10}(300,000/(\mathrm{DF}+1))$$

As explained in section 3.1, *Meta Nodes* are used as a representative for a set of nodes during importance evaluation. Hence, during importance evaluation we do not consider nodes that are connected with any *Meta Node*. To evaluate the importance of a *Meta Node*, we take the number of nodes connected with a *Meta Node* as the *term frequency* for the *Meta Node*.

Step-B: Finding Key Relation- The next step is to find the important relationship between a pair of selected nodes. We use a heuristic in this step. The idea is that the key relationship between the nodes will generally be present in the sentence where they occur together for the first time. If there is no such sentence, then there is probably no important direct relationship between the two nodes, and we ignore the pair. AMRs contain semantic information at the top of the AMR graph.

Table 1: In Example 1, the words **illegal** and **ban** reinforce each others importance but they are not captured by co-reference resolution. We add a *Meta Node* connected to the nodes corresponding to the words **illegal** and **ban**. During importance evaluation, the occurrences of this *Meta Node* will be these occurrences of **illegal** and **ban** and *term frequency* for this *Meta Node* will be 2. Similarly, in the second example both the occurrences of the word **opium** are connected to a new *Meta Node*

1. On 011006 The Citizen newspaper stated that it is **illegal** for South Africans to be involved in mercenary activity or to render foreign military assistance inside or outside of South Africa. The Citizen newspaper stated that the South African Foreign Ministry announced on 011005 that the South African government imposed the mercenary activity **ban** following reports that 1000 Muslims with military training have enlisted to leave South Africa for Afghanistan to fight for the Taliban against the United States.
2. Head of the U.N. drug office Antonio Maria Costa said that Afghanistan has produced so much **opium** in recent years that the Taliban are cutting back poppy cultivation and stockpiling raw opium in an effort to support prices and preserve a major source of financing for the insurgency. Costa said this to reporters last week as the U.N. Drug Office Office prepared to release its latest survey of Afghanistan's **opium** crop.

Table 2: Results on the Proxy report section of the AMR bank. First-half contains the Recall, Precision, and F-1 for the nodes in the generated summary AMR. The second half contains the scores for the final summary generated using state-of-the-art text generator evaluated using the ROUGE metric

	Subgraph extraction			Full pipeline		
Method	Recall	Precision	F_1	R-1	R-2	R-L
Liu et al. (2015)	63.5	54.7	58.7	-	-	-
Unsupervised SAS (Naive co-reference)	**67.9**	57.2	60.3	-	-	-
Unsupervised SAS (Lee et al. (2017) co-reference)	67.0	**58.0**	**60.4**	39.5	16.5	29.0
Unsupervised SAS (Human co-reference)	67.7	60.4	62.4	40.9	16.7	29.5

Thus, in the selected sentence we find a path between the two nodes closest to the root. If one of the selected nodes happens to be a *Meta Node*, the occurrences of the *Meta Node* include all the occurrences of all the nodes that the *Meta Node* represents (Fig. 1).

Step-C: Capture Surrounding Information - The final step in subgraph extraction is to expand around the selected path to capture the surrounding information. We use OpenIE Banko (2009) at this step. The output of the OpenIE system are tuples of the form $(arg; relation; arg)$. The relevant tuples for us are the set of tuples that contain the selected path. As, these tuples contain all the auxiliary information about the relationship that they are describing, selecting a tuple will solve the problem of graph expansion. To capture the maximum amount of auxiliary information we choose the largest tuple among the set of relevant tuples. This ends the process of summary graph extraction. Algorithm 1 provides an overview of the entire algorithm.

3.3 Step 3: Summary Generation

To generate sentences from the extracted AMR graphs we use state of the art AMR to text generator Konstas et al. (2017).

4 Experiments

In table 2 we report results on the test set of the proxy report section of the AMR bank. The table contains results using the human annotated AMRs. We outperform the state-of-the-art in SAS by 1.7% F1 scores in node prediction. Similar to previous methods we use the target summary size to control the length of the output summary.

To evaluate the effectiveness of the method till the summary graph extraction step, we compare the generated summary graph with the gold-standard target summary graph. We report Recall, Precision, and F1 for graph nodes. Finally, to evaluate the effectiveness of the pipeline, we evaluate the performance using ROUGE Lin (2004), and we report ROUGE-1, ROUGE-2, and ROUGE-L.

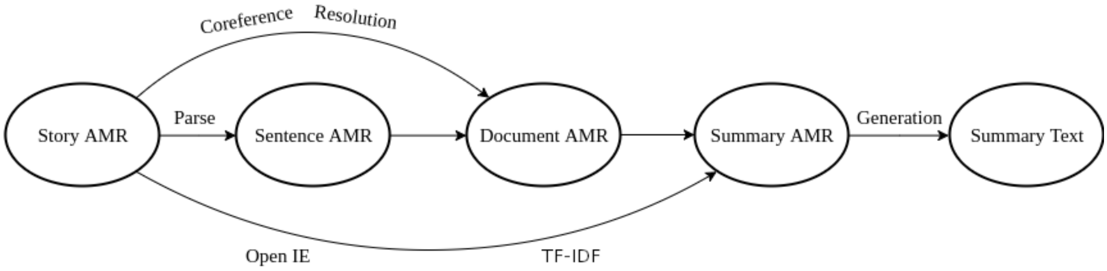


Figure 4: The updated pipeline proposed by us has the following step - AMR Parsing, Co-reference resolution, Open Information Extraction,TF-IDF calculation, Subgraph selection and Text generation

Algorithm 1 Overview of Algorithm

```
Input. Input original text data
Node co-reference resolution and Meta node formation
Node ranking using Tf-idf
Rank nodes
Rank node pairs, using node weights
for Sorted node pairs do
    Select the first co-occurrence sentence
    if no such sentence then
        Continue
    end if
    Find the path closest to root in the sentence
    Find relevant tuples using OpenIE
    Choose the largest tuple containing the path
    if no such tuple then
        Output the selected sentence AMR
    end if
    if summary size > required size then
        Break
    end if
end for
AMR to text conversion
```

As clear from table 2 there is not much difference between the scores when we use naive node resolution and date merging and when we use state-of-art co-reference resolution. To check the impact of co-reference resolution, we also did manual co-reference resolution on the test set which resulted in a further 2% increase in the scores to 62.4%. We suspect that a significant reason for lower performance with state of the art co-reference resolution might be the inability of the system to handle cataphoric references. These references are particularly crucial in news articles where the first occurrence of an entity/event is generally essential.

5 Conclusion and Future Work

In this work, we present a new method to do Semantic Abstractive Summarization (Figure 4). We outperform the previous state-of-the-art methods for SAS by 1.7% and by 3.7% using human co-reference resolution. In the process, we complete the SAS pipeline for the first time showing that SAS can be used to construct high-quality summaries. We also extend the method to construct a document AMR graph from the sentence AMRs using *Meta nodes* which can further be used in some future formalism for Document Meaning Representation.

The work will benefit directly from improvements in each step of the pipeline. Specifically, the advances in co-reference resolution for the near-identity cases might significantly improve the summary quality. We are currently experimenting with bigger text summarization datasets like DUC 2004 and DUC 2006. The hypothesis we used to find the key relations is the main hurdle in extending the work to multi-document summarization as all other steps can be directly applied in multi-document summarization. Using different methods that might be based on supervision to find the key relation is an interest direction for future work.

Acknowledgments

We would like to thank the anonymous ACL reviewers for their insightful comments. We also thank Susmit Wagle and Prerna Bharti of IIT Kanpur for helpful discussions. Vivek Gupta acknowledges support from Microsoft Research Travel Award. We would also like to thank Roam Analytics for sponsoring Shibhansh's ACL SRW travel grant. The opinions stated in this paper are of authors alone.

References

Miguel Almeida and Andre Martins. 2013. Fast and robust compressive summarization with dual decomposition and multi-task learning. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 196–206. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2015. Amr-guidelines.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Martha Palmer, Philipp Koehn, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. Proceedings of Linguistic Annotation Workshop.

Michele Banko. 2009. *Open Information Extraction for the Web*. Ph.D. thesis, Seattle, WA, USA. AAI3370456.

George Chin, Olga A. Kuchar, and Katherine E. Wolf. 2009. Exploring the analytical processes of intelligence analysts. In *InProceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 1120. Association for Computational Linguistics.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. 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*, pages 93–98, San Diego, California. Association for Computational Linguistics.

Hoa Trang Dang and Karolina Owczarzak. 2008. Overview of the tac 2008 update summarization task. In Proceedings of Text Analysis Conference (TAC).

Tobias Falke and Iryna Gurevych. 2017. Bringing structure into summaries: Crowdsourcing a benchmark corpus of concept maps. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2951–2961. Association for Computational Linguistics.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436. Association for Computational Linguistics.

Dan Gillick and Benoit Favre. 2009. A scalable global model for summarization. In Proceedings of the NAACL Workshop on Integer Linear Programming for Natural Langauge Processing.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend.

Youn-ah Kang, Carsten Gorg, and John Stasko. 2011. How can visual analytics assist investigative analysis? design implications from an evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 17(5):570–583.

Kevin Knight, Laura Baranescu, Claire Bonial, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, and Nathan Schneider. 2014. Deft phase 2 amr annotation r1 ldc2015e86. philadelphia: Linguistic data consortium. Abstract meaning representation (AMR) annotation release 1.0 LDC2014T12. Web Download. Philadelphia: Linguistic Data Consortium.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural amr: Sequence-to-sequence models for parsing and generation. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. *CoRR*, abs/1707.07045.

C. Lin. 2004. Rouge: A package for automatic evaluation of summaries. Text Summarization Branches Out, Post-Conference Workshop of ACL 2004. Barcelona, Spain.

Fei Liu, Flanigan Jeffrey, Thomson Sam, Sadeh Norman, and Smith Noah A. 2015. Toward abstractive summarization using semantic representations. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, Denver, Colorado, pages 10771086.

Andre F. T. Martins and Noah A. Smith. 2009. Summarization with a joint model for sentence extraction and compression. In Proceedings of the ACL Workshop on Integer Linear Programming for Natural Language Processing.

Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, pages 557–564, Berlin, Heidelberg. Springer-Verlag.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Sudha Rao, Daniel Marcu, Kevin Knight, and Hal Daumé III. 2017. Biomedical event extraction using abstract meaning representation. In *BioNLP 2017*, pages 126–135, Vancouver, Canada,. Association for Computational Linguistics.

Marta Recasens, Eduard Hovy, and M. Antnia Mart. 2011. Identity, non-identity, and near-identity: Addressing the complexity of coreference. *Lingua*, 121(6):1138 – 1152.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for sentence summarization.

Naomi Saphra and Adam Lopez. 2015. Amrica: an amr inspector for cross-language alignments. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 36–40, Denver, Colorado. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. 2016. Neural headline generation on abstract meaning representation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1054–1059, Austin, Texas. Association for Computational Linguistics.

Lucy Vanderwende, Michele Banko, and Arul Menezes. 2004. Event-centric summary generation. In Proceedings of DUC.

Yannick Versley. 2008. Vagueness and referential ambiguity in a large-scale annotated corpus. *Research on Language and Computation*, 6(3):333–353.

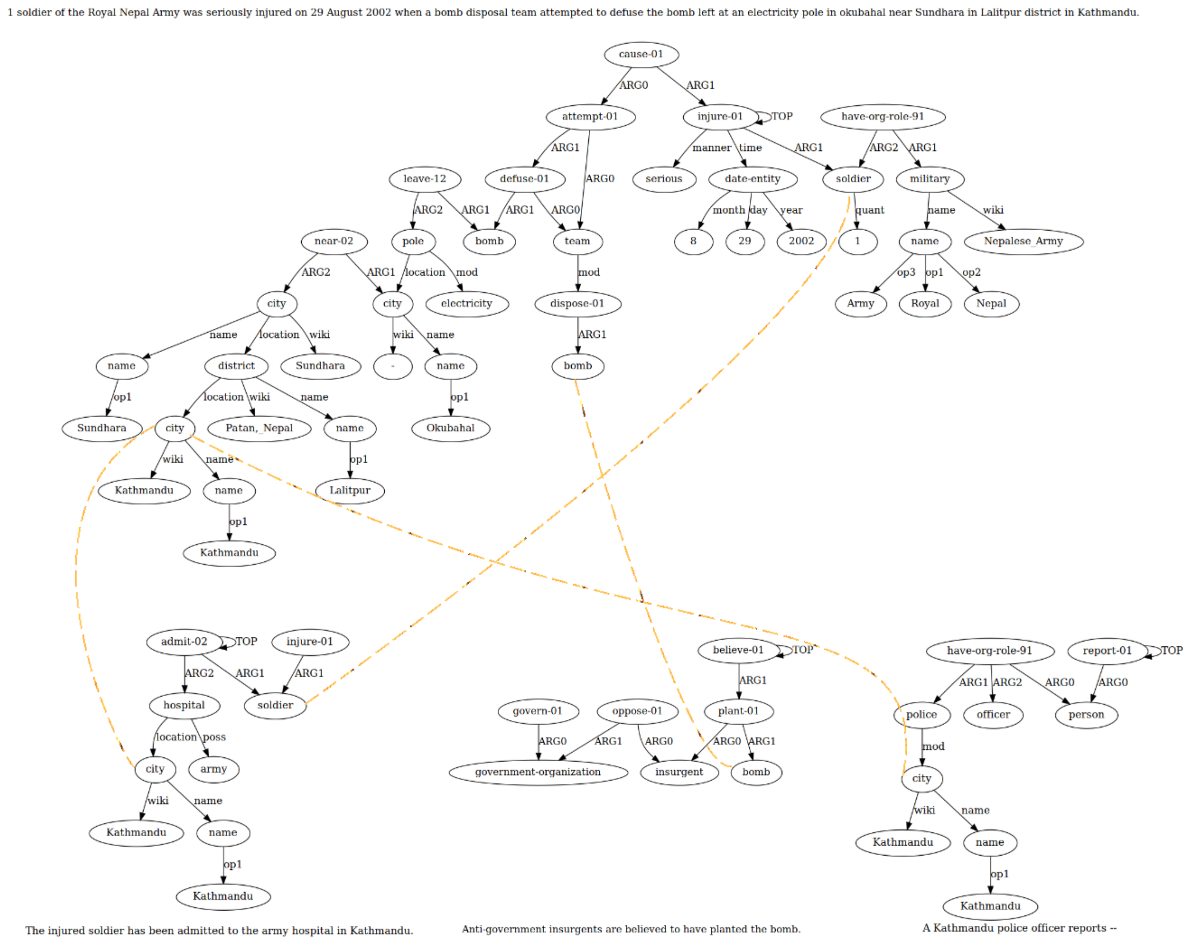
A Example to generate document AMR from sentence AMR

In this appendix, we give an example showing how to generate Document AMR from the sentence AMRs. Consider a short multi-sentence story -

A Kathmandu police officer reports –. 1 soldier of the Royal Nepal Army was seriously injured on 29 August 2002 when a bomb disposal team attempted to defuse the bomb left at an electricity pole in okubahal near Sundhara in Lalitpur district in Kathmandu. Anti-government insurgents are believed to have planted the bomb. The injured soldier has been admitted to the army hospital in Kathmandu.

Figure 5 shows the sentence AMRs of the four sentences of the short story. The nodes that refer to the similar entity have to be merged; the dashed lines connect the nodes to be merged. Figure 6 shows, the generated document AMR from the merger. Fig 6 also shows how large the AMRs of even short stories can become after merging.

If the summarization process were to follow, we would've started by finding the key nodes in the document graph based on $\mathrm{TF}-\mathrm{IDF}$. The Term frequency is the number of incoming edges in the AMR. It is clear from the document AMR that the important nodes based on Term frequency are *Soldier, Bomb,* and *Kathmandu*. Then we use $\mathrm{TF}-\mathrm{IDF}$ to rank among these key nodes, it turns out that the key nodes that the final ranking in decreasing order of importance is *Kathmandu*, *Soldier* and *Bomb*. The next step is to find the key relation, which according to our hypothesis lies in the sentence where they first co-occur, i.e., the second sentence. The exact relation is the highest path. As clear from Figure 5 the path will include the nodes corresponding to the words *Kathmandu*, *Soldier*, *Injured*. And finally in the last step we use the OpenIE system to capture important information surrounding this path.



1 soldier of the Royal Nepal Army was seriously injured on 29 August 2002 when a bomb disposal team attempted to defuse the bomb left at an electricity pole in okubahal near Sundhara in Lalitpur district in Kathmandu.

The injured soldier has been admitted to the army hospital in Kathmandu.

Anti-government insurgents are believed to have planted the bomb.

A Kathmandu police officer reports --

Figure 5: The AMRs of the 4 sentence of the short story. Dashed lines represent the nodes to be merged.

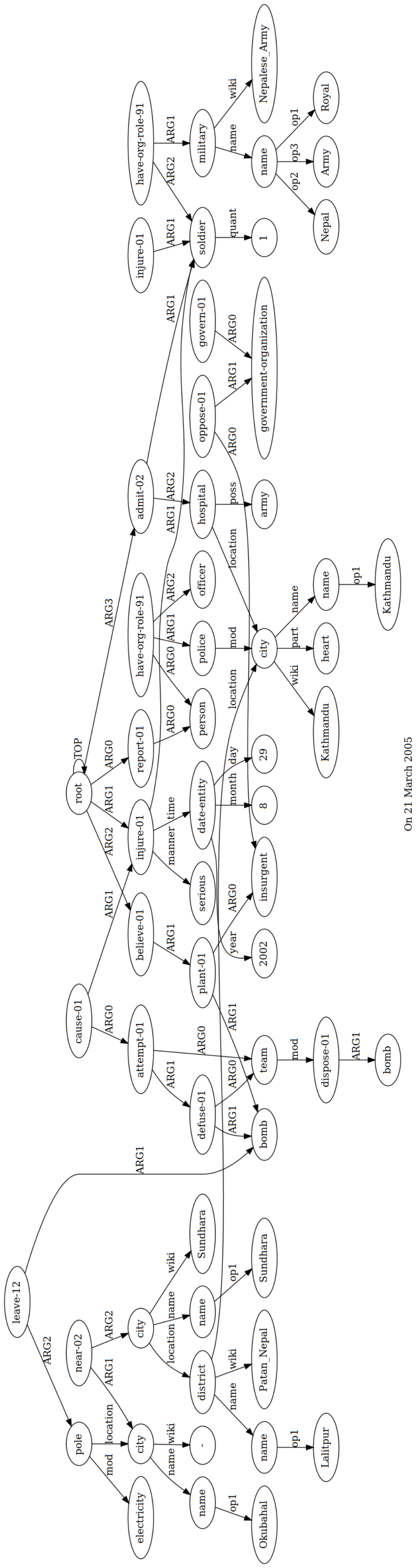


Figure 6: The document AMR generated by merging the four sentence AMRs of the story.