Text summarization using Abstract Meaning Representation

AMR

Amit Nagarkoti
Supervisor: Dr. Harish Karnick

Department of Computer Science and Engineering
Indian Institute of Technology Kanpur

Table of contents

Outline

Introduction

Definition

Text summarization is the process of reducing the size of original document to a much more concise form such that the most relevant facts in the original document are retained.

- Summaries are always lossy.
- Summarization methodologies
  - *Extractive summarization :* extract import words and sentences
  - *Abstractive summarization* **harder** : rephrase, generate similar words

Why Important?

- *user point of view:* Evaluate importance of article.
- *linguistic/scientific/philosophical view:* Solving summarization is equivalent to solving the problem of language understanding.

AMR : welcome to amr

```
(w / welcome-01
  :ARG2 (a / amr))
```

- represent sentence as a DAG
- captures "who is doing what to whom"
- nodes : verb sense *(see-01)*, objects *(boy, marble)*
- edges : relations *(ARG1, ARG0)*

Abstraction in AMR

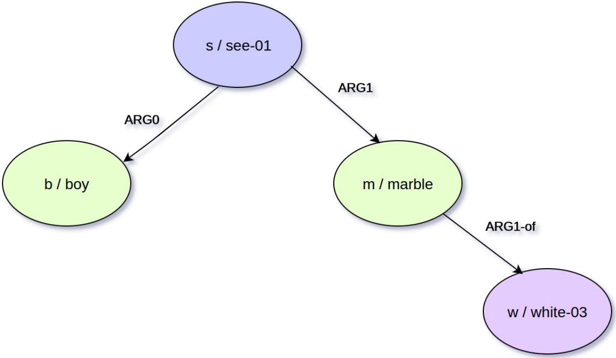

Figure: AMR example

- The boy sees the white marble.
- The boy saw the marble that was white.

Propbank Lexicon

- *white-03* : refers to the color white
  - **ARG1:** thing that is white in color *(marble)*
  - **ARG2:** specific part of ARG1, if also mentioned
- *see-01* : to see or view
  - **ARG0:** viewer *(boy)*
  - **ARG1:** thing viewed *(marble)*
  - **ARG2:** attribute of ARG1, further description *(white - but we have different amr here)*

How to get an AMR?

Use *JAMR* : 84% accuracy for concept node identification.

- gives AMR for a sentence

```
(d / discover-01 | 0
      :ARG0 (r / rope | 0.0)
      :ARG1 (n / noose | 0.1)
      :location (c / campus | 0.2))
```

the noose made of rope was discovered on campus

word-node Alignment

node	**alignment**	**word**
discover-01	6-7—0	discovered
rope	4-5—0.0	rope
noose	1-2—0.1	noose
campus	8-9—0.2	campus

Table: word-node alignment

These alignments will be used to generate summaries from AMRs

Outline

Using AMRs

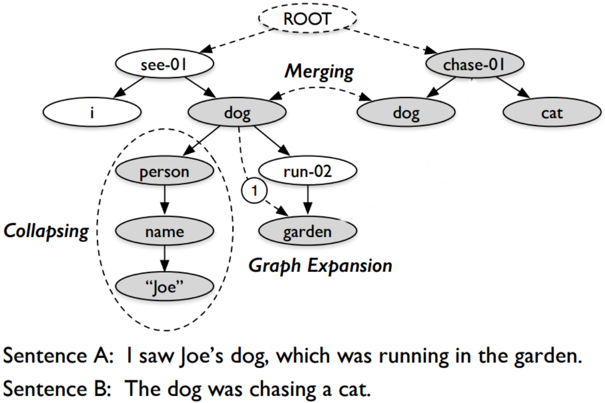

Figure: Steps to generate Document graph from sentence AMR (modified from [Liu et al., 2015])

Document Graph

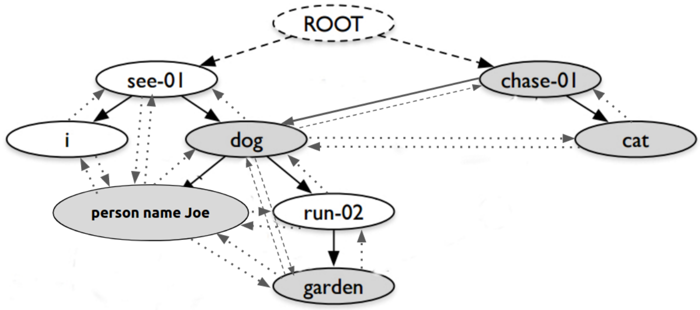

Figure: Dense Document Graph

Finding the summary sub-graph

$$\textit{maximize} \quad \sum_{i=1}^{n} \psi_i \theta^T f(v_i) \tag{1}$$

- here ψ_i is a binary variable indicating node i is selected or not
- $\theta = [\theta_1, \ldots, \theta_m]$ are model parameters
- $f(v_i) = [f_1(v_i), \ldots, f_m(v_i)]$ are node features

Constraints for a valid sub-graph

Figure: $v_i - e_{ij} \geq 0 \quad v_j - e_{ij} \geq 0$

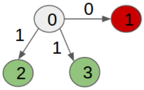

Figure: $\sum_i f_{0i} - v_i = 0$

Figure: $\sum_i f_{ij} - \sum_k f_{jk} - v_j = 0$

$$Ne_{ij} - f_{ij} \geq 0, \forall i, j \quad \textit{sanity constraint}$$

$$\sum_{i,j} e_{ij} \leq L \quad \textit{size constraint}$$

Complete Algorithm

```
for cur_doc in corpus:
 create doc graph
 add ILP constraints
 solve objective to minimize loss
 calculate gradients
 update model parameters
```

LSA for extractive summarization

$$\begin{array}{cc} & \begin{array}{ccc} \phantom{x_{1,1}} & s_j & \phantom{x_{1,n}} \\ & \downarrow & \end{array} \\ \mathbf{t}_i^T \rightarrow & \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{bmatrix} \end{array}$$

Figure: term-sentence matrix

- term vector $t_i^T = [x_{i1} \ \ldots \ x_{in}]$
- sentence vector $s_j^T = [x_{1j} \ \ldots \ x_{mj}]$
- term-sentence matrix can be huge
- no relation between terms
- sparsity

LSA for extractive summarization

$$
\begin{array}{ccccccc}
 & X & & & U & \Sigma & V^T \\
 & \mathbf{s}_j & & & & & \hat{\mathbf{s}}_j \\
 & \downarrow & & & & & \downarrow \\
(\mathbf{t}_i^T) \rightarrow & \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix} & = & (\hat{\mathbf{t}}_i^T) \rightarrow & \begin{bmatrix} \begin{bmatrix} \\ \mathbf{u}_1 \\ \\ \end{bmatrix} \dots \begin{bmatrix} \\ \mathbf{u}_l \\ \\ \end{bmatrix} \end{bmatrix} & \cdot \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{bmatrix} & \cdot \begin{bmatrix} [& \mathbf{v}_1 &] \\ & \vdots & \\ [& \mathbf{v}_l &] \end{bmatrix}
\end{array}
$$

Figure: SVD over term-sentence matrix source:Wikipedia

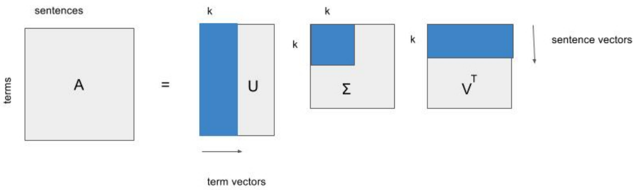

Figure: low rank approximation

LSA for extractive summarization

- Finally a score is calculated for each sentence vector given by

$$S_l = \sqrt{\sum_{i=1}^{n} v_{l,i}^2 \cdot \sigma_i^2} \tag{2}$$

  where S_l is score for sentence l
- choose L sentences with highest scores

Outline

s2s

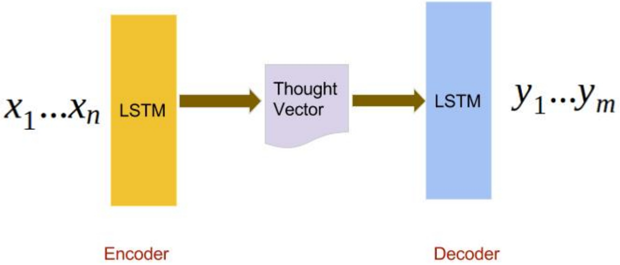

Figure: Sequence to Sequence Model

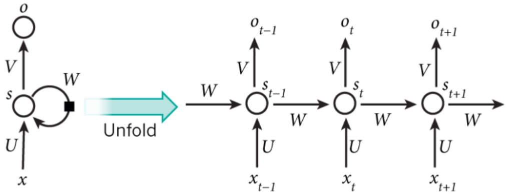

Figure: RNN cell $s_t = f(Ux_t + Ws_{t-1})$ $o_t = softmax(Vs_t)$ source: nature [LeCun et al., 2015]

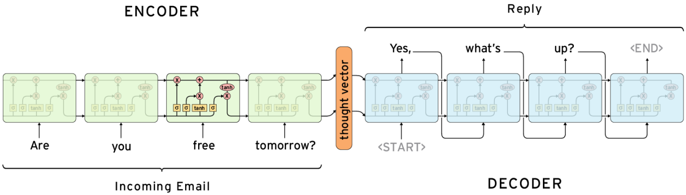

Figure: Chat client using s2s model based on LSTM source: [Christopher,]

S2S continued

During training model tries to minimize the negative log likelihood of the target word

$$loss_D = \sum_{t=1}^{T} -\log(P(w_t)) \tag{3}$$

here w_t is the target word at step t.
Problems in s2s

- slow training with long sequences - *limit sequence lengths*
- limited context for decoder **critical** for s2s
- large vocabularies

Attention to rescue

Attend or Re-visit critical information when making decisions

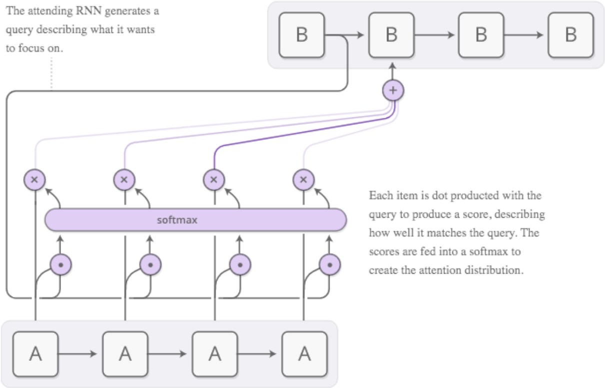

Figure: Attention in RNNs source:distill.pub

Attention complete view

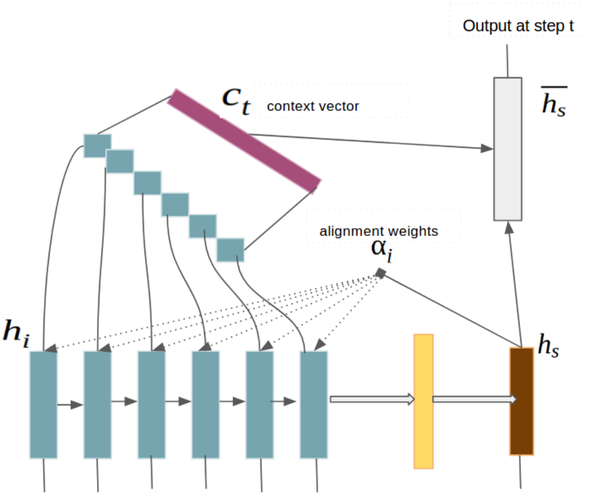

Figure: Attention at step t

Attention Heatmap

$X - axis$ is the input sequence, $Y - axis$ is the generated output

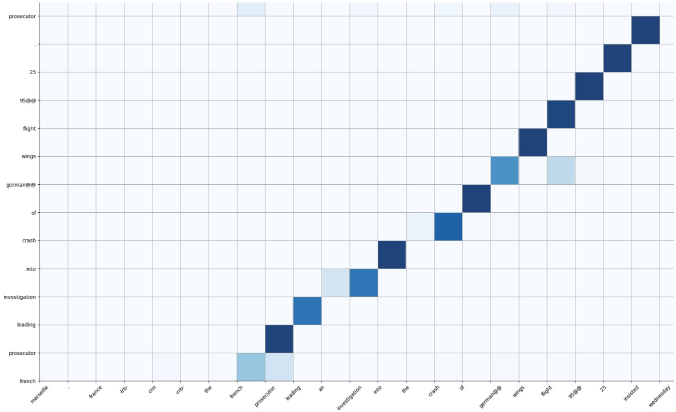

Figure: Attention Heatmap during Decoding

Byte Pair Encoding

Definition

BPE is a compression technique where the most frequent pair of consecutive bytes is replaced by a byte not in the document.

- BPE has been adapted for NMT [Sennrich et al., 2015] using the idea of subword unit.
- "lower" will be represented as "l o w e r @".
- $vocabsize = numOfIterations + numOfChars$.
- BPE merge operations learned from dictionary low, lowest, newer, wider. using 4 merge operations.

r @ - - - - - - - - - - → r@
l o - - - - - - - - - - → lo
lo w - - - - - - - - - - → low
e r@ - - - - - - - - - - → er@

Pointer Generator Model for OOVs

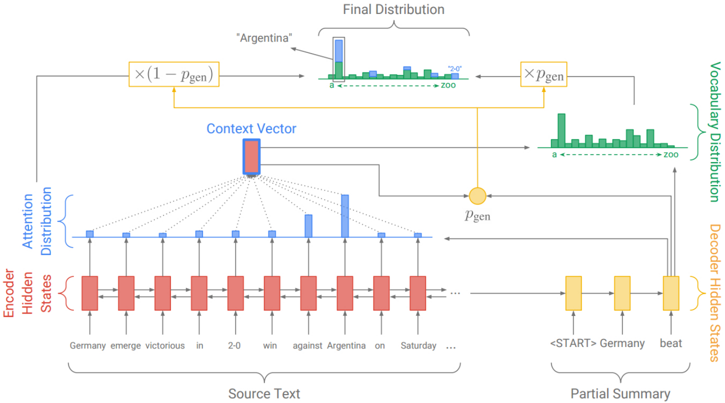

Figure: Pointer Gen model [See et al., 2017]

Coverage

"Coverage" is used in MT to control over and under production of target words.

- Some words may never get enough attention resulting in poor translation/summaries.
- The solution is to use coverage to guided attention [Tu et al., 2016] and [See et al., 2017].
- Accumulate all attention weights and penalize for extra attention.

$$c_t = \sum_{t'=1}^{t-1} \alpha^{t'} \quad \textit{coverage vector} \tag{4}$$

$$covloss_t = \sum_{i=1}^{encsteps} min(\alpha_i, c_i^t) \tag{5}$$

AMR Linearization

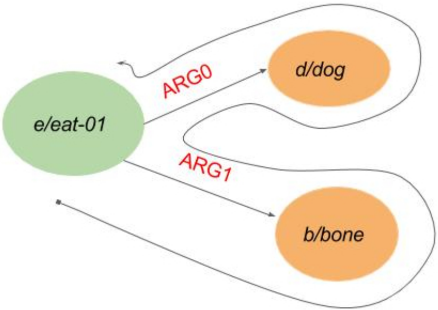

Figure: AMR DFS Traversal gives the linearization -**TOP**-(*eat-01* **ARG1**(*bone*)**ARG1** **ARG0**(*dog*)**ARG0**)-**TOP**-

Data Augmentation/Extension with POS

- POS sequence $p_1 \cdots p_n$ is obtained using the *Stanford Parser*

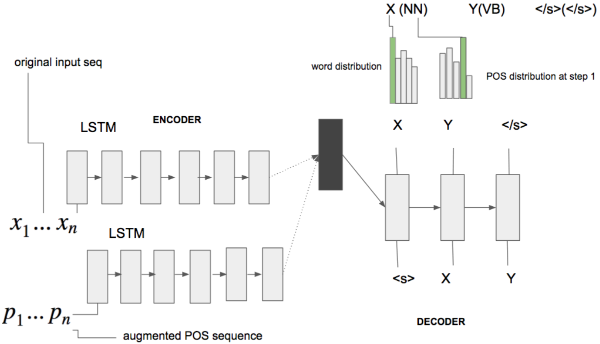

Figure: s2s model generating multiple output distributions

Hierarchical Sequence Encoder for AMRs

- sentence vector S_i is obtained by using attention on word encoder states
- document vector D_i is obtained by using attention on sentence encoder states

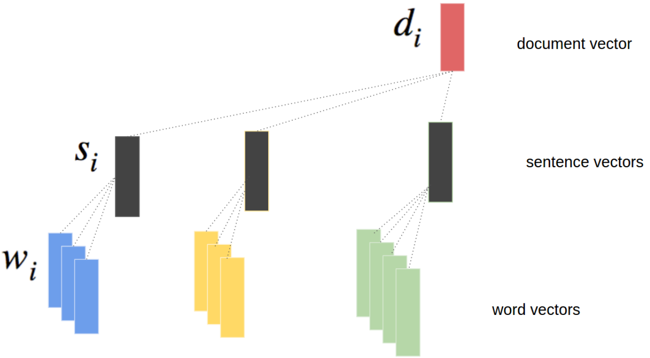

Figure: hierarchical models learn the document vector using level-wise learning

Using Dependency Parsing

"A Dependency Parse of a sentence is tree with labelled edges such that the main verb or the focused noun is the root and edges are the relations between words in the sentence."

- To reduce document size we used a context of size L around the root word, reducing the sentence size to at $max\, 2L + 1$.
- Fig below has "capital" as *root* with $L = 3$ we are able to extract crux of the sentence *i.e* "Delhi is the capital of India".

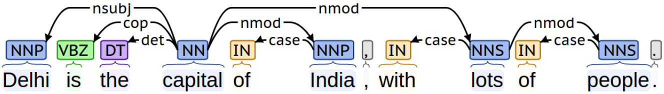

Figure: Dependency Parse :: Delhi is the capital of India , with lots of people.

Outline

Rouge

RougeN-Precision	$\frac{MatchingN-grams}{Count\ candidateN-grams}$
RougeN-Recall	$\frac{MatchingN-grams}{Count\ referenceN-grams}$
RougeN-F1	$\frac{2RougeNre*RougeNpre}{RougeNre+RougeNpre}$
RougeL-Precision	$\frac{lenLCS(ref,can)}{lencandidate}$
RougeL-Recall	$\frac{lenLCS(ref,can)}{lenreference}$
RougeL-F1	$\frac{2RougeLre*RougeLpre}{RougeLre+RougeLpre}$

Table: Rouge Formulas

Examples

- Rouge-N

```
candidate :: the cat was found under the bed
reference :: the cat was under the bed
```

  has recall $\frac{6}{6} = 1$ and precision $\frac{6}{7} = 0.86$

- Rouge-L

```
reference :: police killed the gunman
candidate1 :: police kill the gunman
candidate2 :: the gunman kill police
```

  candidate1 has $RougeL - F1$ of 0.75 and candidate2 has $RougeL - F1$ of 0.5

Dataset Description

We used CNN/Dailymail dataset. The distribution of dataset is as follows

Train	287,226
Test	11,490
Validation	13,368

Table: CNN/Dailymail split

average number of sentences per article	31
average number of sentences per summary	3
average number of words per article	790
average number of words per summary	55
average number of words per article (BPE *vsize* 50k)	818

Table: CNN/Dailymail average stats for Training sets

Results onn CNN/Dailymail

Model	R1-f1	R2-f1	RL-f1
bpe-no-cov	35.39	15.53	32.31
bpe-cov	36.31	15.69	**33.63**
text-no-cov	31.4	11.6	27.2
text-cov	33.19	12.38	28.12
text-200	34.66	16.23	30.96
text-200-cov	37.18	17.41	32.68
pos-full*	35.76	16.9	31.86
pos-full-cov*	**37.96**	**17.71**	33.23

Table: pos-full minimizes loss for word generation and POS tag generation, all models use pointer copying mechanism as default except the bpe model, text-200 uses glove vectors for word vector initialization

AMR S2S

- AMR as augmented data using 25k training examples

Model	R1-f1	R2-f1	RL-f1
aug-no-cov	30.18	11	26.52
aug-cov	**34.53**	**13.22**	**29.21**

Table: AMRs as Augmented Data

- AMR to AMR using s2s

Model	R1-f1	R2-f1	RL-f1
s2s-cnn-1	17.97	6.31	17.35
s2s-cnn-2	25.60	8.31	25.01
s2s-cnn-3	**31.96**	**10.71**	**29.12**

Table: 1: cnn no pointer gen 2: cnn with pointer gen 3: cnn with cov and pointer gen

AMR using Doc Graph

Model	R1-f1
number of edges L <= nodes/2 - 1 (ref gold)	18.59
number of edges L <= nodes/3 -1 (ref gold)	**19.72**
number of edges L <= nodes/4 -1 (ref gold)	19.57
number of edges L <= nodes/2 -1 (ref *generated)	38.05
number of edges L <= nodes/3 -1 (ref *generated)	**44.72**
number of edges L <= nodes/4 -1 (ref *generated)	43.60

Table: Graph based AMR summaries *reference summaries were generated using word alignment in summary graph

Extractive + Abstractive

We used LSA and the dependency parse for the extractive phase

Model	**R1-f1**	**R2-f1**	**RL-f1**
dep-no-cov	25.89	8.72	23.81
dep-cov	30.53	10.26	28.06
lsa-10-no-cov	29.75	11.27	27.3
lsa-10-cov	32.9	12.17	**30.41**
lsa-15-no-cov	31.64	12.34	28.74
lsa-15-cov	**33.22**	**12.86**	30.11

Table: Comparing the mixed approach for summarization, Dependency based method used $L = 7$ as context window value

Sample Outputs

- repetition
  - **Reference:** roseanne barr told the daily beast that she is slowly going blind barr said she has macular degeneration and glaucoma
  - **Generated:** roseanne barr said she is slowly going blind in interview with the daily beast barr said she suffers from macular degeneration and glaucoma [*she suffers from macular degeneration and glaucoma*] [*she suffers from macular degeneration and glaucoma*]
- less abstraction
  - **Reference:** batman superman dawn justice trailer leaked thursday yanked offline film will released march 25 2016 stars ben affleck henry cavill
  - **Generated:** batman superman dawn justice [*leaked handheld camera phone user uploaded handheld camera phone capture trailer spanish*] superman dawn justice will released march 25 2016 [*stars affleck batman henry cavill superman film also stars amy adams*] much costs

Sample Outputs

- facts redundancy
  - **Reference:** coast guard says about 50 people were rescued from mobile bay more than 100 sailboats took part in the dauphin island race an annual event
  - **Generated:** [*sailboats took part in the dauphin island race*] and as many as 50 [*sailboats took part in the dauphin island race*] and as many as 50 people in all were rescued from water [*the coast guard says this is the 57th year*] for the event [*the club says this is the 57th year*] for the event

Conclusion and Future Work

- In this work we walked through some of the techniques for text summarization and suggested some changes to them.
- Extractive methods always degrades the summaries due to lack of complete information but improves on the training times.
- We looked into AMRs and their applicability to the task of summarization and their effectiveness on a smaller data set.
- Finding Global graph embeddings for AMR type structures.
- Using co-referencing in AMRs.
- Data Augmentation without increasing the model complexity.
- Expanding Memory Networks for Summarization.
- Reinforcement Learning for Summarization.
- Better Extraction using Dependency Trees.

Thank You!

```
(t / thank-01
    :ARG1 (y / you))
```

References I

Christopher, O.
Understanding lstm networks.

LeCun, Y., Bengio, Y., and Hinton, G. (2015).
Deep learning.
Nature, 521(7553):436–444.
Insight.

Liu, F., Flanigan, J., Thomson, S., Sadeh, N., and Smith, N. A. (2015).
Toward abstractive summarization using semantic representations.

See, A., Liu, P. J., and Manning, C. D. (2017).
Get to the point: Summarization with pointer-generator networks.
arXiv preprint arXiv:1704.04368.

References II

Sennrich, R., Haddow, B., and Birch, A. (2015).
Neural machine translation of rare words with subword units.
arXiv preprint arXiv:1508.07909.

Tu, Z., Lu, Z., Liu, Y., Liu, X., and Li, H. (2016).
Modeling coverage for neural machine translation.
arXiv preprint arXiv:1601.04811.