Automatic
Text Summarization

Introduction

- Text produced from one or more texts , that conveys important information of the original text/texts
- There are two different approaches for automatic text summarization

Extractive Summarization :

approaches select passages from the source text, then arrange them to form a summary. You might think of these approaches as like a highlighter.

Abstractive Summarization :

approaches use natural language generation techniques to write novel sentences. By the same analogy, these approaches are like a pen.

The great majority of existing approaches to automatic summarization are extractive – mostly because it is much easier to *select* text than it is to *generate* text from scratch.

Extractive Summarization

- extractive summarization techniques produce summaries by choosing a subset of the sentences in the original text.
- In order to fetch important sentences from the text , we can
  1. Construct an intermediate representation expressing main aspects of the text
  2. Score the sentences based on representation
  3. Select a summary comprising of required length.

- Luhn’s method(1958) - count based
  - Ignore Stop words
  - Determine Top Words: The most often occuring words in the document are counted.
  - Select Top Words: A small number of the top words are selected to be used for scoring.
  - Select Top Sentences: Sentences are scored according to how many of the top words they contain.

Extractive Summarization

Topic Representation Approaches

* Used frequency thresholds to locate the descriptive words and represent the topic

SumBasic

1) $$g(S_j) = \frac{\sum_{w_i \in S_j} P(w_i)}{|\{w_i \mid w_i \in S_j\}|}$$

$P(w_i) = f(w)/N$

word probability

2) Best scoring sentence that contain highest prob word

3) $P_{new}(w_i) = P_{old}(w_i) \cdot P_{old}(w_i)$ → reduce the probabil of already present word

→ After stopword removal

* TFIDF = $q(w) = f_d(w) * \log \frac{|D|}{f_D(w)}$

$|D|$ → Total no. of docum

$f_D(w)$ → # documents containing word w.

$f_d(w)$ → term frequency of w in document d

Latent Semantic Analysis

- Documents are made of topics and topics are made of words .
- A good Summarization should explain the most important topics
- Create a word sentence matrix representing occurrences/tf-idf/log-entropy and use SVD to

a mixture of topics

term – sentence matrix → each topic is mixture of words

a_{ij} – weight of word i in sentence j

$$A = \underset{n \times m}{U} \Sigma \underset{m \times m}{V^T}$$

U – term topic matrix

$D = \Sigma V^T$ topic – sentence matrix

↳ d_{ij} → wt of topic i in sentence j

$g(s_i) = \sqrt{\sum_{i=1}^{m} d_{ij}^2}$ → wt of sentence i

Latent Semantic Analysis

- Create a word sentence matrix representing occurrences/tf-idf/log-entropy and use SVD to factorize the matrix .

book	0.15	-0.27	0.04
dads	0.24	0.38	-0.09
dummies	0.13	-0.17	0.07
estate	0.18	0.19	0.45
guide	0.22	0.09	-0.46
investing	0.74	-0.21	0.21
market	0.18	-0.3	-0.28
real	0.18	0.19	0.45
rich	0.36	0.59	-0.34
stock	0.25	-0.42	-0.28
value	0.12	-0.14	0.23

*

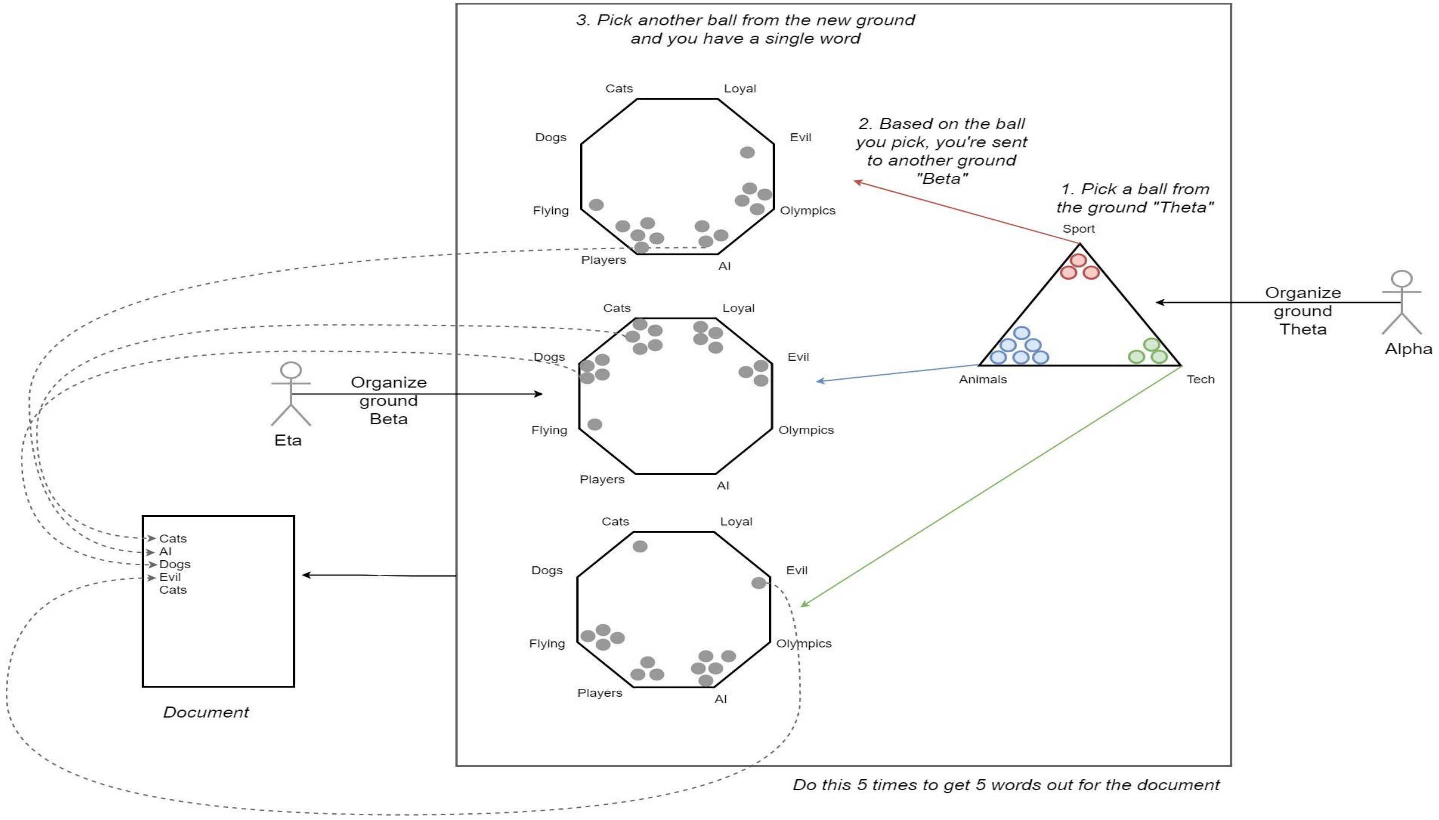
3.91	0	0
0	2.61	0
0	0	2

*

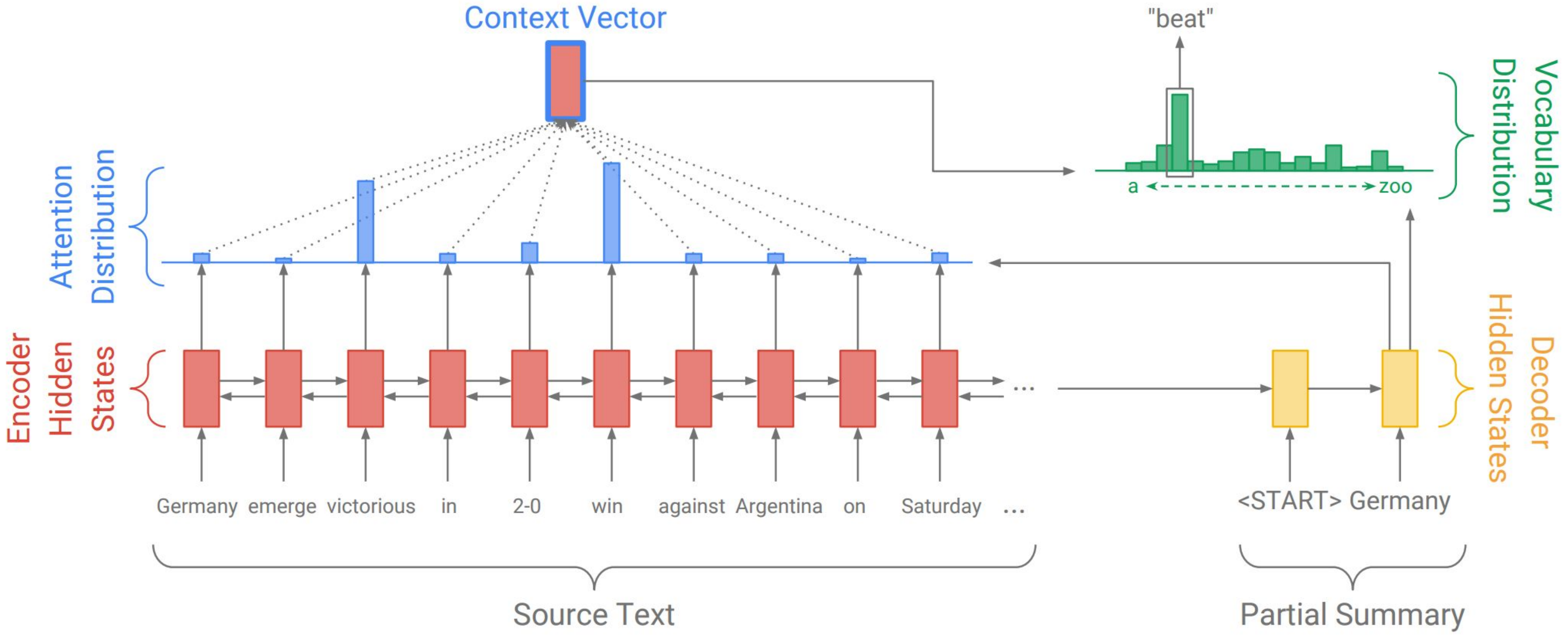
T1	T2	T3	T4	T5	T6	T7	T8	T9
0.35	0.22	0.34	0.26	0.22	0.49	0.28	0.29	0.44
-0.32	-0.15	-0.46	-0.24	-0.14	0.55	0.07	-0.31	0.44
-0.41	0.14	-0.16	0.25	0.22	-0.51	0.55	0	0.34

- Drawbacks ::
  - LSA depends heavily on SVD which is computationally intensive and hard to update as new documents appear.
  - There is no probabilistic interpretation to scores

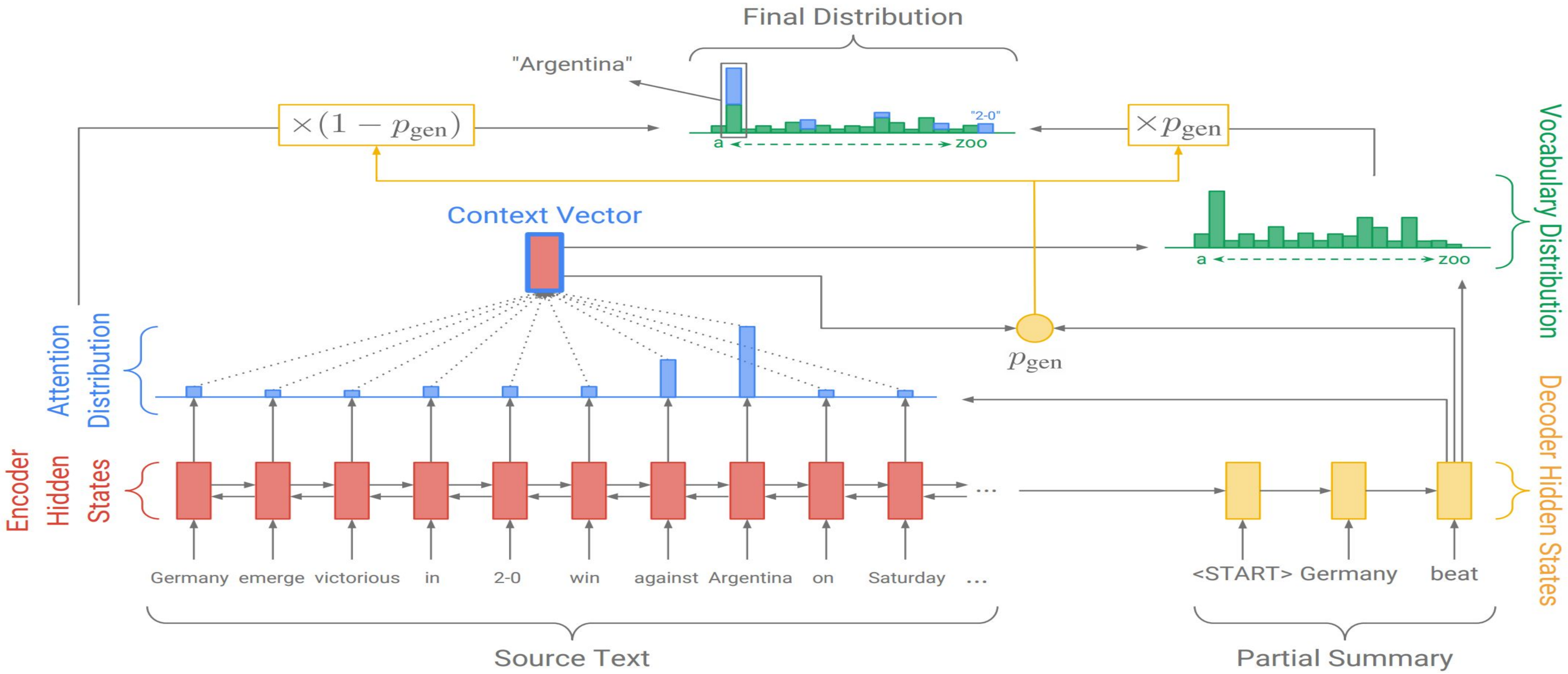
Latent Dirichlet Allocation



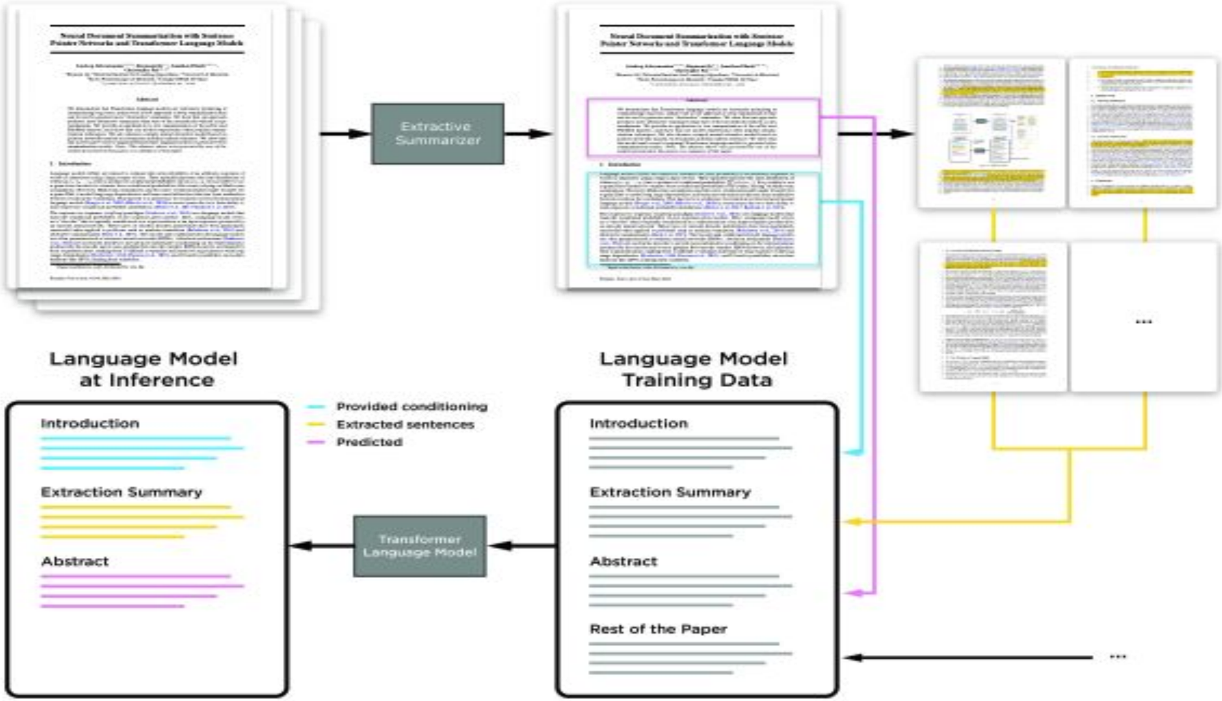
Abstractive Summarization - seq2seq



Pointer-Generator networks



Using Extractive to generate Abstractive



ROUGE

Rouge score

Measures recall, how much human - summary appeared in machine reference summaries.

Rouge recall & precision

Rouge recall → $\frac{\text{\# of overlapping words}}{\text{\# of words in reference}}$

Precision → $\frac{\text{\# overlapping words}}{\text{\# words in system}}$

ROUGE - N → n-grams instead of words

ROUGE - L → matches Longest common subsequence

ROUGE - S → skip-gram coocurrence

RL in seq2seq

- The model's aim is to output the reference summary, so we define a cross entropy loss between the target and the produced word. But this approach is fundamentally flawed.
- There are various ways in which the document can be effectively summarized. The reference summary is just one of those possible ways.
- So, the model's aim shouldn't be just restricted to outputting only the reference summary. There should be some scope for variations in the summary.
- This is the idea behind using Reinforcement learning in Summarization.