Ben-Gurion University of the Negev
The Faculty of Natural Sciences
The Department of Computer Science

Question Answering as an Automatic Summarization Evaluation Metric

Matan Eyal

Thesis submitted in partial fulfillment of the requirements
for the Master of Sciences degree

Under the supervision of Prof. Michael Elhadad

September 2018

Ben-Gurion University of the Negev
The Faculty of Natural Sciences
The Department of Computer Science

Question Answering as an Automatic Summarization Evaluation Metric

Matan Eyal

Thesis submitted in partial fulfillment of the requirements
for the Master of Sciences degree

Under the supervision of Prof. Michael Elhdad

Signature of student: _____ Date: _____
Signature of supervisor: _____ Date: _____
Signature of chairperson of the
committee for graduate studies: _____ Date: _____

September 2018

אוניברסיטת בן־גוריון בנגב
הפקולטה למדעי הטבע
המחלקה למדעי המחשב

הערכת סיכומים בעזרת יכולת מענה על שאלות

מתן איל

חיבור לשם קבלת התואר "מגיסטר" בפקולטה למדעי הטבע

בהנחיית פרופסור מיכאל אלחדד

ספטמבר 2018

תקציר

עבודות מהתקופה האחרונה בנושא סיכום אוטומטי וייצור כותרות, מתמקדות במיטוב שיטת אומדן הנקראת ROUGE עבור מסדי נתונים שונים. בעבודה זו אנו מציגים שיטת אומדן חלופית עבור משימה זו. שיטה זו, מנצלת התקדמויות במשימת הבנת הנקרא האוטומטית, על מנת לכמת את היכולת לענות, מתוך הסיכום, על שאלות הנוגעות לאושיות מרכזיות. תחילה, אנו מנתחים את חוזק שיטת אומדן זו על ידי השוואתה עם שיטות אומדן ידניות. לאחר מכן אנו מציגים מודל אבסטרקטי עצבי המביא למרב את שיטת האומדן המוצעת, ובזאת גם משפרים את תוצאות ROUGE לכדי תחרויות עם התוצאות העדכניות ביותר.

Abstract

Recent work in the field of automatic summarization and headline generation focuses on maximizing ROUGE scores for various news datasets. We present an alternative, extrinsic, evaluation metric for this task, ***A**nswering **P**erformance for **E**valuation of **S**ummaries*. APES utilizes recent progress in the field of reading-comprehension to quantify the ability of a summary to answer a set of manually created questions regarding central entities in the source article. We first analyze the strength of this metric by comparing it to known manual evaluation metrics. We then present an end-to-end neural abstractive model that maximizes APES, while increasing ROUGE scores to competitive results.

Acknowledgements

First and foremost I would like to thank my adviser, Prop. Michael Elhadad, for guiding me and providing consultation in every step in the road, for challenging me to experiment and research new fields and for the good word when needed. I would also like to thank all my colleagues: Tal Baumel, Ben Eyal, Dan Schulman, Avi Hayoun, Jumana Nassour-Kassis, Dr. Yael Netzer, and Dr. Meni Adler, I couldn't have done it without you. Lastly I would like to thank my family, my wife, parents, brother and sister for always believing in me.

Contents

List of Figures

List of Tables

1 Introduction

Summarizing documents is a challenging task for computational linguistics: It consists of reading and analyzing a source document and producing a shorter version which captures the key information from the source. A key challenge in the task lies in its evaluation: different people will produce different summaries from the same source, all perfectly legitimate and equally valid. How can we assess the quality of a summary produced by an algorithm? Matching the summary produced by a specific human author is not sufficient. In the past 20 years, the problem of evaluating summarization algorithms has settled on an automatic method called ROUGE, which captures the lexical similarity between a generated summary and a collection of typically about 3 to 5 reference summaries produced for the same source document by human authors through a number of easy to compute metrics (ROUGE-1, ROUGE-2, ROUGE-SU4, ROUGE-L measured with Precision, Recall or F-measure). A series of experiments has confirmed a strong correlation between the ROUGE scores for an algorithm and the assessment produced by human readers for small-scale datasets of about 50 source documents in the DUC and TAC series of shared tasks.

Recently, the dominant paradigm of summarization techniques, which

consists of extractive methods where full sentences from the source documents are selected and ordered into the target summary has been challenged by emerging deep-learning-based abstractive methods. In contrast to the simpler extraction methods, abstractive methods re-generate the target document word by word based on a reading of the source document. Algorithms of this type must be trained end to end on large datasets containing pairs of (source document, summary). Existing datasets of this type contain a single reference summary (which often, is not even generated as a summary but as a headline for news articles). Under these conditions, it is natural to question whether the existing ROUGE metrics capture summary quality in the same way as in the settings in which ROUGE was developed and validated in previous research: there is a single reference summary as opposed to multiple, the generated summary is generated word by word as opposed to extractive, the reference summary is not generated as a summary but as a headline which may have other communicative function besides conveying the key information in the source document.

In this work, we address the question of assessing the quality of abstractive summarization algorithms trained on large datasets of single reference summary. We develop a new automatic evaluation method, which we call APES, which is more semantic in nature than ROUGE. We find that APES complements ROUGE by providing additional focus on capturing key information from the source document which is not directly measured by ROUGE. A combination of ROUGE and APES provides improved guidance to generate informative and readable summaries.

In addition, we present an end-to-end neural abstractive model which

optimizes this metric and by that also increasing ROUGE scores. We evaluate our algorithm on standard large-scale datasets of News documents and assess the correlation between manual assessment of summary quality and our APES metrics and with ROUGE on the smaller traditional TAC datasets, where manual assessment is available.

In this introduction, We present the fundamental components of automatic summarization methods and lay the motivation that led us to suggest our novel evaluation metric ***A**nswering **P**erformance for **E**valuating **S**ummaries* (APES). We cover the subjects of automatic summarization and summarization evaluation metrics. We then present some of the challenges and open questions in the field of automatic summarization and finally we present the principle underlying our automatic evaluation metric ***A**nswering **P**erformance for **E**valuating **S**ummaries*.

1.1 Automatic Summarization

The task of automatic text summarization aims to produce a concise version of a source document while preserving its central information. While this objective is well defined, the method in which the summarizer decided to produce the summary can vary in the following independent ways.

Abstractive/Extractive Summaries

One way of dividing summarization models is into *extractive* and *abstractive* models. In extractive summarization, summaries are created by select-

ing a collection of key sentences from the source document (*e.g.*, Nallapati et al. [23], Narayan et al. [25]). Abstractive summarization on the other hand, aims to rephrase and compress the input text in order to create the summary. Progress in sequence-to-sequence models [38] has led to recent success in abstractive summarization models. Current models [2, 24, 29, 36] made various adjustments to the sequence-to-sequence model in order to improve ROUGE [19] scores.

Informative/Indicative Summaries

Approaches to summarizing a source document can also be distinguished as either *informative* or *indicative*. An informative summary is meant to replace the need for the original source document, as it aims to hold all the central information from the original text. Indicative summaries on the other hand, intend to help the reader decide whether to continue and read the original text. All models mentioned in this work are informative summaries.

Single/Multi Document Summaries

Another dimension in summarization is the type of input given to the system. The input can either have a single document or multi documents. In the single document case, the central information which we want to include in our generated summary can be hard to identify. In the multi-document case, a set of documents concerning the same topic are given. Here, the central information might be easier to determine as that information should be part of mostly all the source documents. All models

mentioned above received a single document as input.

1.2 Summarization Evaluation Metrics

One of the challenges in the field of text summarization is finding a method to evaluate a proposed summary. In many other fields in Natural Language Processing (*e.g.*, Part of Speech Tagging, Syntactic Parsing), given an input, there is a single output the model is required to produce. On the other hand, in the field of text summarization, given a source document, a number of summaries can be defined as satisfactory proposals. For this reason, the field of summarization evaluation is in constant search for various automatic and manual evaluation metrics.

Evaluation of the quality of summaries can be assessed by comparing source and target text and ideal summaries using various similarity measures and text quality metrics. Such evaluation methods are called *intrinsic* and they include most currently used techniques such as ROUGE [19] and the Pyramid method [26]. Alternatively, *extrinsic evaluation methods* perform task-based assessment: they verify how people perform a task using the summaries instead of using the source text itself like suggested in Steinberger and Ježek [37, Sect 3.4]. Such tasks may include text categorization, information retrieval or question answering [18].

Manual Evaluation

Manual evaluation metrics, being done by humans, are believed to be more accurate over automatic evaluations. While humans are indeed more

reliable, this advantage is also a disadvantage as manual labor is more expensive in time and effort.

One of the mostly used manual metric is the Pyramid method [26]. The Pyramid method is a manual method to analyze multiple human-made summaries into "Summary Content Units" (SCUs) and assign importance weight to each SCU. An SCU is a collection of short snippets of text capturing a single predicate - for example, *'Google acquired Waze'* and *'Waze Inc. was acquired by Internet search giant Google'* would belong to the same SCU. Different summaries are scored by assessing the extent to which they convey SCUs according to their respective weights. This method requires multiple human-made summaries and manual intervention to detect SCUs in reference summaries and in generated summaries. In particular, detecting the presence of an SCU in text requires the capability to recognize paraphrases.

Automatic Evaluation

While manual evaluation metrics are indeed believed to be more accurate, lack of consistency between different evaluators and high cost in time and effort pushed the research community to develop automatic evaluation metrics. The most popular metrics are the different ROUGE metrics [19].

ROUGE [19], or "Recall-Oriented Understudy for Gisting Evaluation" is a group of intrinsic automatic evaluation metrics. The most common metrics from the ROUGE group are ROUGE-N, that evaluate summaries by counting the overlap of N-grams (1,2 are common choices of N). Two additional metrics from this family are ROUGE-L, which evaluates a sum-

mary by identifying the Longest Common Subsequence (LCS) between the produced summary and the source documents, and ROUGE-SU, which uses skip-bigram plus unigram-based co-occurrence statistics. ROUGE has achieved its status as the most common method for summaries evaluation by showing extremely high correlation to manual evaluation methods (*e.g.*, the Pyramid method [26]). In 2011, the TAC's AESOP shared task [27] attempted at developing automatic summarization metrics. It established ROUGE as a strong baseline, and confirmed the correlation of ROUGE with manual evaluation to an extent not matched by other proposed automatic methods.

1.3 Challenges in Summarization

The task of summarization draws some similarities to other sequence-to-sequence tasks (like Neural Machine Translation) but a number of challenges are specific to this task:

Central Information Identification

One characteristic of a summarization model is to distinguish between central and support information in the source text. While some of the challenges automatic models encounter are easy for human domain experts (*e.g.*, translation or part of speech tagging), this objective can be challenging for humans as well.

Redundancy

When information is covered in multiple source documents, the likelihood of this information being pivotal to the summary increases. The summarization model should, however, avoid repeating this pivotal information more than once. The task of avoiding redundancy is mostly relevant for multi-document summarization model, but it remains an issue even for single document summarization, because variants of the same information can be developed in multiple paragraphs in slightly different forms.

Coherence

When producing a summary, entities references should be included from the original source document without losing coherence. The risk of losing coherence is particularly high in extractive summarization, because when a sentence is extracted from its source context, it may lose the link to previously mentioned entities. The same risk exists naturally for abstractive summarization, because the text generator must learn to produce references in a logical order.

1.4 *Answering Performance for Evaluating Summaries* Metric

While it has been shown that ROUGE is correlated to Pyramid, Louis and Nenkova [20] showed that this summary level correlation decreases significantly when only a single reference is given. In contrast to the smaller

manually curated DUC datasets used in the past, more recent large-scale summarization and headline generation datasets (*CNN/Daily Mail* [13], Gigaword [12], New York Times [34]) provide only a single reference summary for each source document. In this work, we introduce a new automatic evaluation metric more suitable for such single reference news article datasets.

We define APES, ***A**nswering **P**erformance for **E**valuation of **S**ummaries*, a new metric for automatically evaluating summarization systems by querying summaries with a set of questions central to the input document (see Fig. 1.1).

APES evaluates the quality of a summary by measuring how many questions can be answered correctly by reading a summary, from a set of questions generated on the basis of the source document, and aiming at capturing the key information in the source document. When the summary is informative and focused, one should be able to answer many questions just by reading it.

Naturally, in order to use APES automatically, we need to have access to (1) a set of questions for each source document as shown in Fig.1.1; (2) a method to automatically answer questions by reading a generated summary.

Recent progress in the field of automatic question answering (QA) makes our evaluation scheme possible. Chen et al. [3] analyze a model that can reach 'ceiling performance' for the question answering task on the *CNN/Daily Mail* dataset. To measure the APES metric of a candidate summary, we run such an automatic QA system on the summary for each of a set of refer-

See et al. [36]'s Summary: bolton will offer new contracts to emile heskey , 37 , eidur gudjohnsen , 36 , and adam bogdan , 27 . heskey and gudjohnsen joined on short-term deals in december . eidur gudjohnsen has scored five times in the championship .
Baseline Model Summary: bolton will offer new contracts to emile heskey , 37 , eidur gudjohnsen , 36 , and goalkeeper adam bogdan , 27 . heskey and gudjohnsen joined on short-term deals in december , and have helped neil lennon 's side steer clear of relegation . eidur gudjohnsen has scored five times in the championship , as well as once in the cup this season .
Our Model: bolton will offer new contracts to emile heskey , 37 , eidur gudjohnsen , 36 , and goalkeeper adam bogdan , 27 . heskey joined on short-term deals in december , and have helped neil lennon 's side steer clear of relegation . eidur gudjohnsen has scored five times in the championship , as well as once in the cup this season . lennon has also fined midfielders barry bannan and neil danns two weeks wages this week . both players have apologised to lennon .
Questions: goalkeeper _____ also rewarded with new contract; answer: adam bogdan _____ and @entity20 both fined by club after drinking incident; answer: barry bannan @entity18 and _____ both fined by club after drinking incident; answer: neil danns

Figure 1.1: Example 3083 from the test set: our model's summary answers correctly to all three available fill-in-the-blank type questions for this article. Only one question was answered correctly by the other models.

ence questions associated to the source document. The APES metric is the percentage of questions which were answered correctly over the whole dataset.

Given a dataset of source documents / reference summaries, APES requires that we produce a set of focus questions to identify key information in each document. We demonstrate different ways to achieve this goal in this work for different types of datasets - in particular in the News genre which is prevalent in the field of summarization.

1.5 Contributions

Our contributions in this work are: (1) We first present APES as a summarization evaluation metric ; (2) We show that APES is correlated to Pyramid [26] and Responsiveness [8] manual metrics; (3) We then present a new abstractive model which maximizes APES by increasing attention to salient entities, while increasing ROUGE to competitive level. We make the pretrained models and our code available online.

In the rest of the thesis, we present the starting points of the work: recent neural methods for sequence to sequence transformation of text. We then review related previous work in summarization evaluation and abstractive summarization methods. We formulate our research question, present the details of the APES evaluation method and how it correlates with manual evaluation methods and with ROUGE on TAC dataset. We then introduce APESO, an abstractive summarization model trained end to end to optimize the APES score. We describe experiments to assess the quality of APESO: the details of the datasets used, experiments and results. We conclude with an error analysis, identifying the differences between APESO and baseline models, and weak spots of the model. We conclude with contributions and future work.

2 Sequence-to-sequence Methods

Summarization is a function from one sequence of words to another where the input is the original source document of which we want to summaries, and the output is the summary. The output should hold specific characteristics, *e.g.*, being shorter than the input document while preserving its salient information. Similarly, Neural Machine Translation is too a function from a sequence of words to another sequence where the output is the translation of the first sequence to a required language. Any such functions are called sequence-to-sequence (or seq2seq) methods. In the following sections, we describe the basic building blocks for such methods.

2.1 Word Embeddings

When examining differences between human and machine in the context of natural language processing, the way we encode words differ. Historically, words, in the context of natural language processing, are encoded as a one hot encoding, i.e., a vocabulary-sized vector where all-but-one of its elements are 0, and a single element, the word index, is 1. For exam-

ple, if the index corresponding to the word "dog" is 200, the corresponding vector is $((0)_{199}1(0)_{vocab\_size-200})$. Similarly, assuming the index corresponding to the word "hound" is 28543, we receive the following vector $((0)_{28542}1(0)_{vocab\_size-28543})$. Notice that while the meaning for these words is very similar, the corresponding vector representations (or word embeddings) are orthogonal. Moreover, the vector representing the words "dog" is similarly distant to the word "hound" as it is to the word "phone" for example.

An additional issue with the one-hot representation of words is the size of the vectors, as each vector is the size of the vocabulary while only representing a single index.

While a thorough understanding of human encoding of words is beyond the scope of this work, the field of natural language processing aims to learn a word representation where semantically similar words should be mapped to nearby points in the embedded vector space.

For these reasons better representation of words has been studied: a representation that aims to capture semantic understanding of a word in a dense continuous smaller dimensional space.

2.1.1 Word2Vec

The most commonly used word embeddings are pre-trained embeddings developed in Mikolov et al. [22]. This method, also known as word2vec, is an end-to-end statistical model designed to learn word-embeddings. In their words, Mikolov et al. [22] suggested two unsupervised methods for better representation, the Continuous bag-of-words (CBOW) and the Skip-

Gram methods. These methods are unsupervised, as there is no "true" observed representation for words. Instead, the authors trained the learned representation to solve the following task: given a *window* of words of size $2j$, $w_{i-j}, \cdots, w_{i-1}, w_{i+1}, \cdots, w_{i+j}$, we aim to predict the center word, w_i. This corresponds to the CBOW method. Similarly, for the Skip-Gram method, the authors suggested that given a center word w_i the model should predict the context words $w_{i-j}, \cdots, w_{i-1}, w_{i+1}, \cdots, w_{i+j}$ for some j. The authors developed an efficient algorithm to learn vector representations for words given a very large corpus of sentences (about 100 billions word occurrences). Their algorithm includes Hierarchical Softmax, Negative Sampling and Sub-sampling of Frequent Words and is demonstrably scalable.

In many neural models of sequence to sequence transformation, word are encoded using pre-trained word2vec representations on a large corpus. Alternatively, for some tasks, it is more efficient to train the word embeddings as part of the end to end task optimization.

2.2 Recurrent Neural Networks

Recurrent Neural Networks, or RNNs, are a class of recursive neural networks designed to capture the sequential property of language. While non-recurrent networks receive inputs independently of each other, the sequential nature of language requires a network that holds a state representing all previous information passed beforehand and iteratively proceeds to the next part of the sequence, as words are read one by one, left to right. For example, we would like a recurrent model that receives "I am

your father...you are my..." will be able to predict the word "son". In order to predict the next token, our network should "remember" a suitable representation of the sentence prefix to derive the correct prediction.

The basic component of the RNN is called the RNN cell, and its original implementation, sometimes called Elman-RNN, is defined by the following equations:

$$h_t = tanh(w_{ih}x_t + b_{ih} + w_{hh}h_{t-1} + b_{hh}) \tag{2.1}$$

Where w_{ih}, w_{hh}, b_{ih} and b_{hh} are learnable parameters. h_t is the hidden state at time t, x_t is the input at time t. Other implementations, with other activation function are possible.

Variant concrete implementations have been suggested over the years to implement RNNs. The most successful for linguistic data in recent years are the LSTM (Long Short Term Memory Recurrent Networks) Hochreiter and Schmidhuber [15] and the GRU (Gated Recurrent Units Networks) Chung et al. [7]. In our work, we used LSTMs Hochreiter and Schmidhuber [15] as our RNN cell.

2.3 Sequence to Sequence Model

Sequence-to-sequence models, or Encoder-Decoder models, were first proposed by Sutskever et al. [38]. This model consists of two components: an encoder and a decoder.

The encoder is an RNN. Preferred implementation in recent years is

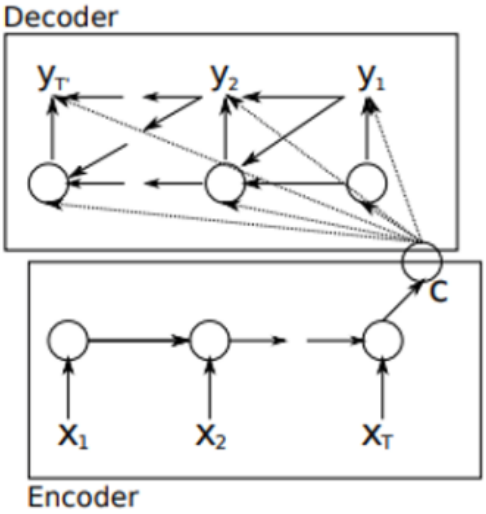

Figure 2.1: Sequence to sequence paradigm as illustrated in [5]

defining the encoder as a Bidirectional Long-Short Term Memory (BiLSTM) [15] with one or more layers. The encoder receives as input the source document, $\mathbf{x} = (x_1, \cdots, x_{T_x})$, where each word x_i is represented as a word embedding vector, and produces hidden states $\mathbf{h} = (h_1, \cdots, h_{T_x})$. We then define c as the latent representation of the source document.

$$c = q(\{h_1, \cdots, h_{T_x}\}) \tag{2.2}$$

Historically, the input is represented by the last hidden state, that is $q(\{h_1, \cdots, h_{T_x}\}) = h_{T_x}$. In other words, the encoder encodes the input tokens, in our case, the source document, into a fixed length vector representation. Fig 2.1 demonstrates the Encoder-Decoder paradigm in its basic form.

The Decoder is an RNN that decodes the encoder's final representation

into a new token, y_t, at each time step, in our case, the summary. We would like to maximize our probability of generating the correct token, with respect to the reference output, at every timestep. Formally, we want to maximize:

$$\prod_{i=1}^{T} p(y_i|\{y_1, \cdots, y_{i-1}\}, c) \tag{2.3}$$

Notice that we are given the target words from previous time steps (*Teacher Forcing*). We model this conditional probability function with the following linear layers:

$$p(\mathbf{y}|\{y_1, \cdots, y_{i-1}\}, c) = \text{softmax}(V'(V[c||s_t] + b) + b') \tag{2.4}$$

Where V, V', b, b' are learnable parameters. The output of the softmax in the RHS of the equation is a probability distribution in the size of the vocabulary, essentially returning a probability distribution over the next word we want to generate.

One might rightfully claim that the bottleneck for this sequence-to-sequence model is the single fixed length vector responsible to encode all the source document.

2.4 Attention Mechanism

In order to solve this bottleneck in representation Bahdanau et al. [1] suggested the Attentional Encoder-Decoder model for the task of Neural Machine Translation (NMT). The authors expanded the Encoder-Decoder model

with the following method.

The authors used a bidirectional RNN for the Encoder, so that the current hidden state represents not only the preceding words, but also the following words. Given a forward hidden state and a backward hidden state, $\vec{h_i}$ and $\overleftarrow{h_i}$ respectively, we denote h_i to be the concatenation of the two, $[\vec{h_i}, \overleftarrow{h_i}]$.

The authors most influential contribution is to relieve the encoder from the 'burden' of encoding all of the source document representation in a fixed-size vector. Instead the authors suggested to compute a context vector at each decoding step that encodes the latent representation of the source document, with respect to the current decoder state. The structure of this so-called *soft attention* model is displayed in Fig. 2.2 and modeled in the following way:

$$\begin{aligned} c_i &= \sum_{j=1}^{T_x} a_{ij} h_j \\ a_{ij} &= \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \\ e_{ij} &= v_a^T \tanh(W_s s_i + W_h h_j + b_a) \end{aligned} \tag{2.5}$$

Let s_i be the decoder hidden state at timestep i.

$$s_i = f(c_{i-1}, s_{i-1}) \tag{2.6}$$

Where f is an LSTM cell and y_i is the target output at decoding step i. h_j is the encoder hidden states.

c_i, known as the *context vector* is a fixed-size representation of the source

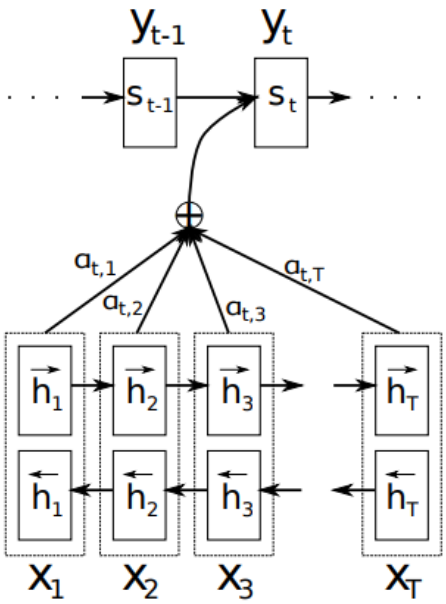

Figure 2.2: Sequence to sequence paradigm with attention as illustrated in [1]

with respect to the decoder state. Notice that c_i is a linear combination of the weights, denoted by a_{ij}, and the encoder hidden states h_j. In other words, high weights reflect high influence of the corresponding hidden states on the context vector. a_i is a probability distribution over the input document decided by the energies e_i where each e_{ij} is a function of the current decoder state and the j-th encoder hidden state.

In this implementation, the encoder no longer needs to represent all the source document in a single vector. Instead, a different representation of the document is derived at each decoding step with respect to the current decoder state. Intuitively, the normalized weights a_{ij} represent an alignment between input and output tokens: when a_{ij} is high, the token x_j from the source has a high impact on the selection of the token y_i in the gener-

ated sequence. Fig.2.3 illustrates this interpretation of attention weights as alignment in an attentional sequence to sequence translation model from English to French.

Given the computed context representation c_i, for each decoding step i, we produce P_v^i, a probability distribution over the vocabulary to select the next generated token y_i.

$$\begin{aligned} o_i &= V[c_t||s_t] + b \\ P_v^t &= \text{softmax}(V' o_t + b') \end{aligned} \tag{2.7}$$

Where V, b, V', b' are learnable parameters. The shared vocabulary is the vocabulary created by the K most frequent words in the trainset from the source *and* the target documents.

Fig. 2.4 illustrates encoding and decoding flow for an attentional network for the task of abstractive summarization as portrayed by See et al. [36]

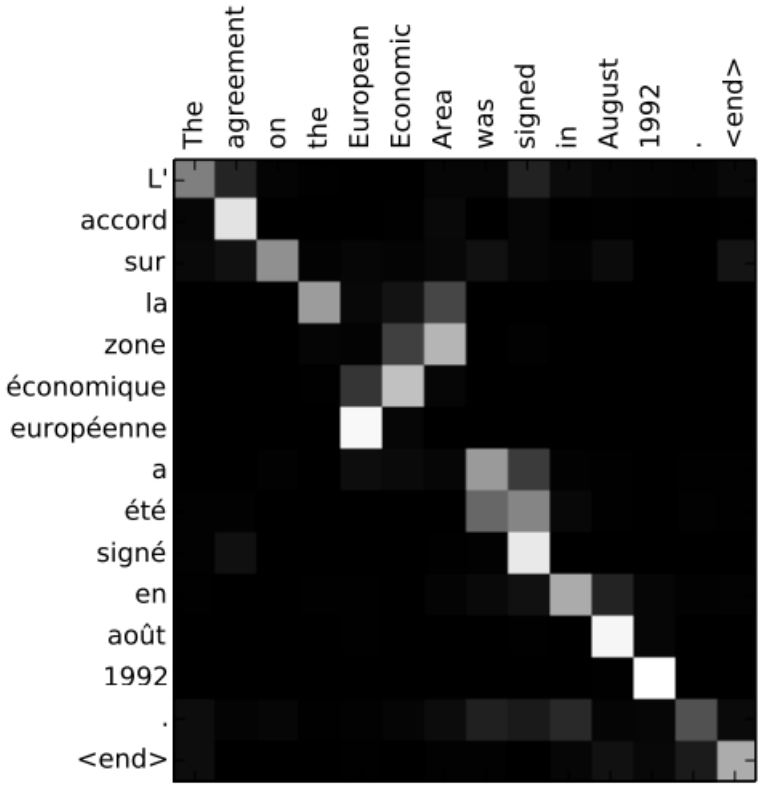

Figure 2.3: Translating from the x-axis language (English) to the y-axis language (French). Each cell shows the weight a_{ij} for the i-th target word where high values correspond to White and low values correspond to Black (From [1])

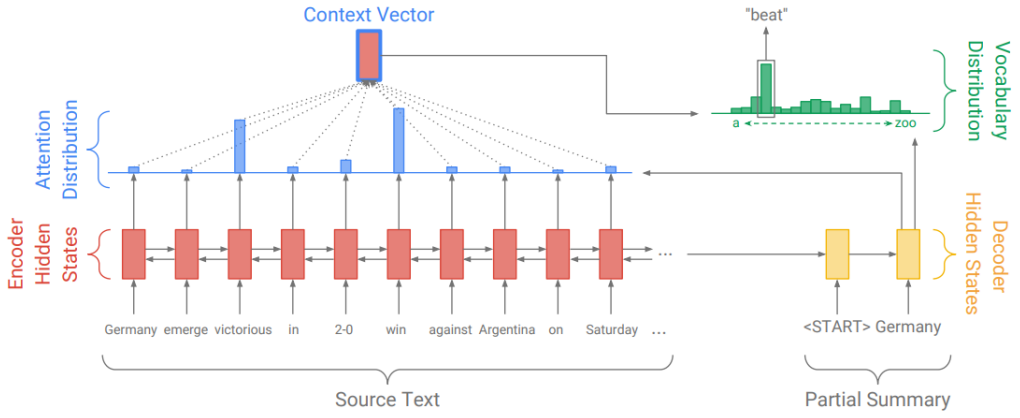

Figure 2.4: Attentional Encoder-Decoder as presented in See et al. [36]

3 Previous Work

3.1 Evaluation Methods

Automatic evaluation metrics of summarization methods can be categorized into either *intrinsic* or *extrinsic* metrics. Intrinsic metrics measure a summary's quality by measuring its similarity to a manually produced target gold summary or by inspecting properties of the summary. Examples of such metrics include ROUGE [19], Basic Elements [16] and Pyramid [26]. Alternatively, extrinsic metrics test the ability of a summary to support performing related tasks, and compare the performance of humans or systems when completing a task that requires understanding the source document [37]. Such extrinsic tasks may include text categorization, information retrieval, question answering [18] or assessing the relevance of a document to a query [14].

ROUGE, or "Recall-Oriented Understudy for Gisting Evaluation" [19] refers to a set of automatic intrinsic metrics for evaluating automatic summaries. ROUGE-N scores a candidate summary by counting the number of N-gram overlaps between the automatic summary and the reference summaries. Other notable metrics from this family are ROUGE-L, where

scores are given by the Longest Common Subsequence (LCS) between the suggested and reference documents, and ROUGE-SU4, that uses skip-bigram, a more flexible method for computing the overlap of bigrams.

The Pyramid method [26] is a manual evaluation metric that analyzes multiple human-made summaries into "Summary Content Units" (SCUs) and assigns importance weights to each SCU. Different summaries are scored by assessing the extent to which they convey SCUs according to their respective weights. This method, in its most reliable form, requires multiple human-made summaries alongside manual intervention to detect SCUs in source and target documents. The Basic Elements method [16], an automated procedure for finding short fragments of content, has been suggested to automate a method related to Pyramid. Like Pyramid, this method requires multiple human-made gold summaries, making this method expensive in time and cost as well. Responsiveness [8], another manual metric is a measure of overall quality combining both content selection, like Pyramid, and linguistic quality. Both Pyramid and Responsiveness are the standard manual approaches for content evaluation of summaries.

Other relevant quantities for summaries quality assessment include: readability (or fluency), grammaticality, coherence and structure, focus, referential clarity and non-redundancy. Although some automatic methods were suggested as summarization evaluation metrics [41], [39], these metrics are commonly assessed manually, and, therefore, rarely reported as part of experiments.

Our proposed evaluation method, APES, attempts to capture the capability of a summary to enable readers to answer questions – similar to the

manual task originally discussed in Jing et al. [18] and recently reported in Narayan et al. [25]. Our contribution consists in automating this method and assessing the feasibility of the resulting approximation.

3.2 Neural Methods for Abstractive and Extractive Summarization

The first paper to use an end-to-end neural network for the abstractive summarization task was Rush et al. [33]: this work is based on sequence-to-sequence models [38] augmented with an attentional decoder [1] as explained in 2.4. Nallapati et al. [24] was the first to tackle the headline generation problem using Hermann et al. [13] dataset adopted for the summarization task. This work also adopted the work from Vinyals et al. [42] adding an additional layer of a pointer mechanism to the network. The pointer mechanism determines in the decoder whether to generate words based on the current state of the decoder or instead, to copy words verbatim from the source. This decision is critical in the News domain typically used for summarization benchmarks, because it covers the case of rare named entities, which are not processed naturally by a standard encoder-decoder architecture. See et al. [36] followed the work of Nallapati et al. [24] and added an additional layer, the coverage mechanism, in order to reduce repetitions in decoding time. Paulus et al. [29] introduce intra-attention in order to attend over both the input and previously generated outputs. The authors also present a hybrid learning objective designed to maximize ROUGE scores using Reinforcement Learning.

Nallapati et al. [23] adopted a neural architecture to the task of extractive summarization. In this model, the text encoder produces a representation of the source sentences which is used to classify each sentence as 'kept' or 'discarded' from the target summary. This model improved ROUGE scores over existing extractive and abstractive models. Narayan et al. [25] suggested a variant neural extractive summarization model to optimize ROUGE scores, where they model the task as a sentence ranking problem, show that cross-entropy training is not sufficient at optimizing ROUGE, and propose a reinforcement learning objective. As discussed above, this work also reports human question answering evaluation.

All these papers have been evaluated using ROUGE, and all, except for Rush et al. [33], used *CNN/Daily Mail* as their main headline generation dataset. In this work, we introduce a new abstractive model which starts from the encoder-decoder with attention and pointer mechanism, and design a new loss to improve the APES objective, while maintaining high ROUGE performance.

3.3 Analysis of Related Work in Neural Methods for Abstractive Summarization

All the papers presented below share a common baseline infrastructure, consisting of the component described in Section 2.

3.3.1 Abstractive Sentence Summarization with Attentive Recurrent Neural Networks

Model

Chopra et al. [6] suggest to train the attention mechanism proposed in Bahdanau et al. [1] on sentence-summary pairs. The paper reached state-of-the-art (SOTA) results on multiple datasets comparing to SOTA models at the time the paper was written (ABS+ [33]).

The proposed Attentional Encoder-Decoder model, called in the paper Recurrent Attentive Summarizer (RAS), is trained while minimizing the negative conditional log likelihood (NLL) of $P(\mathbf{y}|\mathbf{x};\theta)$, where θ are the learnable parameters.

Let

$$P(\mathbf{y}|\mathbf{x};\theta) = \prod_{t=1}^{N} p(y_t|\{y_1,\cdots,y_{t-1}\},x;\theta). \tag{3.1}$$

be the language model induced by the article-summary pairs. We minimize the corresponding conditional log likelihood of $P(\mathbf{y}|\mathbf{x};\theta)$:

$$L = -\sum_{i=1}^{B}\sum_{t=1}^{N} \log P(y_t^i|\{y_1^i,\cdots,y_{t-1}^i\},x^i;\theta) \tag{3.2}$$

Where B is the number of documents in this batch. At test time, generation is done using beam search such that $P(y|\mathbf{x})$ is maximized.

Dataset and Training

To optimize the loss function the authors used Stochastic Gradient Decent (SGD) with mini-batchs of size 32 with different RNN cells, trying

both standard RNN and LSTM. Hyper-parameters were chosen using grid search over the validation set with the final architectures being a 2 layers RNN/LSTM where hidden size is 512 and learning rate is 0.5.

Results

The model is evaluated on a randomly selected sample of 2,000 pairs from the Gigaword corpus. The authors also show results for 500 pairs from the DUC-2004 dataset. Evaluation is based on three measurements, ROUGE-1, ROUGE-2 and ROUGE-L.

Gigaword dataset: After choosing the model that showed best results in terms of perplexity on the validation set, the authors compute the F1-score of the different ROUGE metrics. Comparing ABS+ [33], and the new models, RAS-RNN (called in the paper RAS-Elman) and RAS-LSTM, the best performing model is RAS-RNN with 33.78, 15.97 and 31.15 ROUGE-1, ROUGE-2 and ROUGE-L scores respectively beating the ABS+ model by more than 4 points for each of the ROUGE metrics.

DUC-2004: For this dataset the best model is again RAS-RNN beating the SOTA by about 0.2 points for each of the ROUGE metrics. This time recall-only ROUGE results were reported.

3.3.2 Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond

The contribution of Nallapati et al. [24] is three-fold. First, similarly to Chopra et al. [6], the authors implement the attention encoder-decoder

RNN model suggested for NMT in Bahdanau et al. [1] and claim that the model, without any further changes, outperforms the SOTA model at the time the paper was written. Secondly, it's important to note that despite the similarities, abstractive summarization has different set of difficulties than NMT. So the authors suggest different novel models that provide improvement to the baseline model performance. Lastly, a new dataset is proposed for the abstractive summarization task. A dataset that becomes the main dataset in this field.

Suggested Models

In this paper, 4 models were suggested.

Encoder Decoder RNN with Attention and Large Vocabulary Trick: The baseline model in this paper is the Attentional Encoder-Decoder suggested in Bahdanau et al. [1] but this time the encoder is a bidirectional GRU-RNN, as suggested in Chung et al. [7]. The decoder is a unidirectional GRU-RNN with an attention mechanism. The authors also experimented with the Large Vocabulary 'Trick' (LVT) as suggested in Jean et al. [17], that is restricting the decoder's vocabulary to the union of all the words in the source document and the most frequent words in the vocabulary. The aim is to reduce the computational bottleneck in the softmax layer of the decoder: instead of "softmaxing" over all the vocabulary size, the decoder is sampling from a smaller set of words.

Capturing Keywords using Feature-rich Encoder: One of the challenges in summarization is understanding the main entities and concepts in a document. In order to find these attributes the authors suggest adding ad-

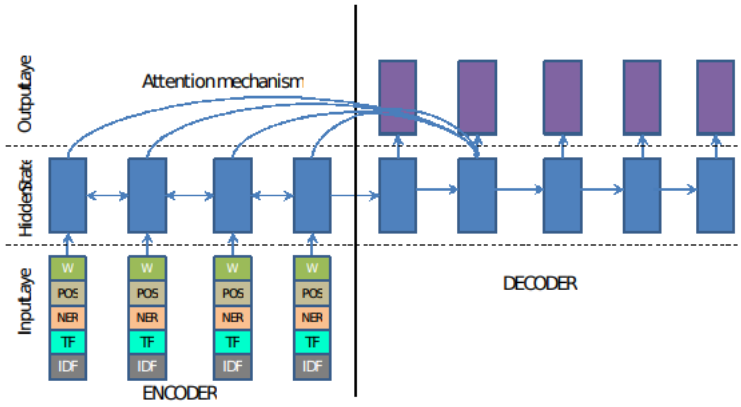

Figure 3.1: Feature-rich encoder model as proposed in Nallapati et al. [24]

ditional information to each word embedding. The information that was added is parts-of-speech tags, named-entity tags, and TF-IDF statistics. An illustration of the system is displayed in Fig. 3.1

Modeling Rare/Unseen Words using Switching Generator-Pointer: The authors introduce a pointer networks, as suggested in Vinyals et al. [42], to deal with rare or unseen words at generation time. In pointer networks, we learn a generation-switch: when on, we generate like in previous models (sample a word from the vocabulary in the decoder, using the softmax distribution), but when the switch is off, we instead point to an index in the source document and copy the corresponding word into our summary. This will be explained in greater detain in the next section. An illustration of the system is displayed in Fig. 3.2

Capturing Hierarchical Document Structure with Hierarchical Attention: Dealing with long documents is another problem in summarization. In this paper the authors suggest, in addition to identifying the keywords in the

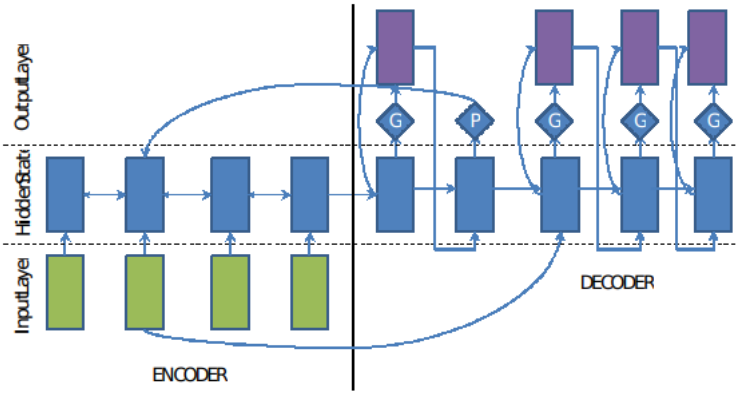

Figure 3.2: Switching generator/pointer model as proposed in Nallapati et al. [24]

document, to find the key sentences. They suggest an hierarchical model where using two BIdirectional RNNs, one like in previous models, and one at the sentence level where the attention mechanism operates at both levels simultaneously. An illustration of the system is displayed in Fig. 3.3.2

In later stages of the paper, after realizing the model tends to repeat itself, another model suggested is using temporal attention, as suggested in Sankaran et al. [35]. The attention mechanism that was introduced discourages the model from repeating itself by remembering past attention weights. See et al. [36] research this topic in great detail. Beside using the intra-attention as used here in and Paulus et al. [29], another option researched in great detail is the coverage-mechanism as explained in See et al. [36].

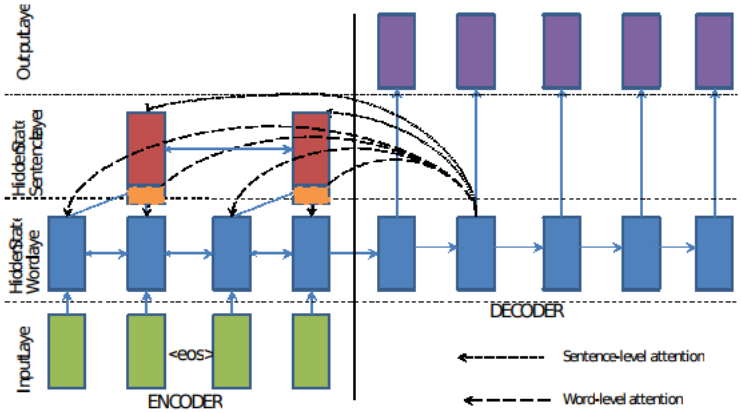

Figure 3.3: Hierarchical encoder with hierarchical attention model Nallapati et al. [24]

Dataset and Training

Gigaword Corpus. The authors used 200-dimentions embedding vectors from the Mikolov et al. [22] pretrained vectors uploaded online, with fine-tuning the vectors during training. The hidden state size is 400 and batch size of 50 while sorting according to the documents length to speed up training. They used LVT of size 2000 for all of their models, and it cut down training time per epoch by nearly three times. They also used beam search of size 5 in generation time.

DUC Corpus. Because of the relative small sizes of DUC-2003 and DUC-2004, both Rush et al. [33] and Chopra et al. [6] trained their model on Gigaword dataset and evaluated on the DUC dataset. In this paper they simply run the models as is on the DUC dataset.

CNN/Daily Mail Corpus. This dataset one of the major contributions of

this paper, the authors used the dataset curated for the reading comprehension task by Hermann et al. [13] and repurposed it for the summarization task. This will be explained in greater detains in following chapters in this work.

For this dataset the authors had different vocabulary for the encoder and the decoder limited 150k and 60k respectively. Documents were cut short if they passed maximum length of 800 for the source document and 100 for the summary. Embedding is 100-dimensional word2vec pretrained of this dataset and hidden state size of 200. Because of the relative big size of the dataset training took between 7 and 18 days for different models on a single K-40 GPU.

Results

Gigaword Corpus. Evaluation is done on a randomly selected 2000 examples each for the validation and the test sets. The authors compare their model to the recent SOTA results (ABS+ and RAS-RNN) showing statistical significant improvement in all ROUGE metrics. In order to produce comparable results, the authors used the same test set as was used in Rush et al. [33]. The best model was the baseline model introduced in this paper where LVT size is 5k and is trained to predict the first sentence.

DUC Corpus. The authors trained only 2 models, the baseline model with different LVT sizes, 2k and 5k. Comparing these models' results to recent SOTA models we see that the suggested models beat other models in 2 of the 3 ROUGE metrics.

CNN/Daily Mail Corpus. As this is the first time this corpus is evaluated,

suggested models can't be compared to recent SOTA. Although comparing the different suggested models it appears the best model is the baseline model with 2k LVT size with temporal attention.

3.3.3 Get To The Point: Summarization with Pointer-Generator Networks

See et al. [36] contributions are the novel *coverage mechanism* and a change in the *pointer mechanism* comparing to the mechanism used in Nallapati et al. [24]. The coverage mechanism was introduced in order to solve the repetition problem that arises in sequence-to-sequence models.

Suggested Models

See et al. [36] baseline model is exactly like Nallapati et al. [24] baseline model. To this, the authors added a pointer-generator network [42] as in the Nallapati et al. [24], however, implemented somewhat differently: In Nallapati et al. [24] the switch is trained to be active only for out-of-vocabulary (OOV) words or named entities, whereas in this model the switch isn't restricted to these scenario solely. That is, in every decoding step we calculate p_{gen}, the *generation probability* where we learn a soft-switch of indicating of whether want to generate, per usual, or copy a token from the source document.

$$p_{gen} = \sigma(W_c c_t + W_s s_t + W_x x_t + b_{ptr}) \tag{3.3}$$

Where W_c, W_s, W_x and b_{ptr} are learnable parameters and σ is the sig-

moid function. Given p_{gen} we sample w from the extended vocabulary, $P(w)$, that is the concatenation of the fixed vocabulary of the K-most frequent words in the train documents, and all the words in the source document. Define $P(w)$ in the following way

$$P(w) = p_{gen}P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} a_i^t \tag{3.4}$$

A figure corresponding to this model is presented in 3.4

Another key contribution introduced in this paper is the coverage mechanism. It is similar to the one suggested in Tu et al. [40] for the NMT problem. See et al. [36] suggests to define the coverage vector as the sum of attention distributions over all previous decoder timesteps.

$$cov^t = \sum_{t'=0}^{t-1} a^{t'} \tag{3.5}$$

Then, this vector is fed as an input to the attention layer in order to ensure that the attention mechanism ”knows” its previous distributions in the following way:

$$e_{ij} = v_a^T \tanh(W_s s_i + W_h h_j + W_{conv} cov_i^t + b_a) \tag{3.6}$$

Additionally, the loss function is modified so we will penalize repeatedly attending the same locations in the following way:

$$loss_t = -\log P(w_t^*) + \lambda \sum_i min(a_i^t, conv_i^t) \tag{3.7}$$

where λ is a hyperparamater.

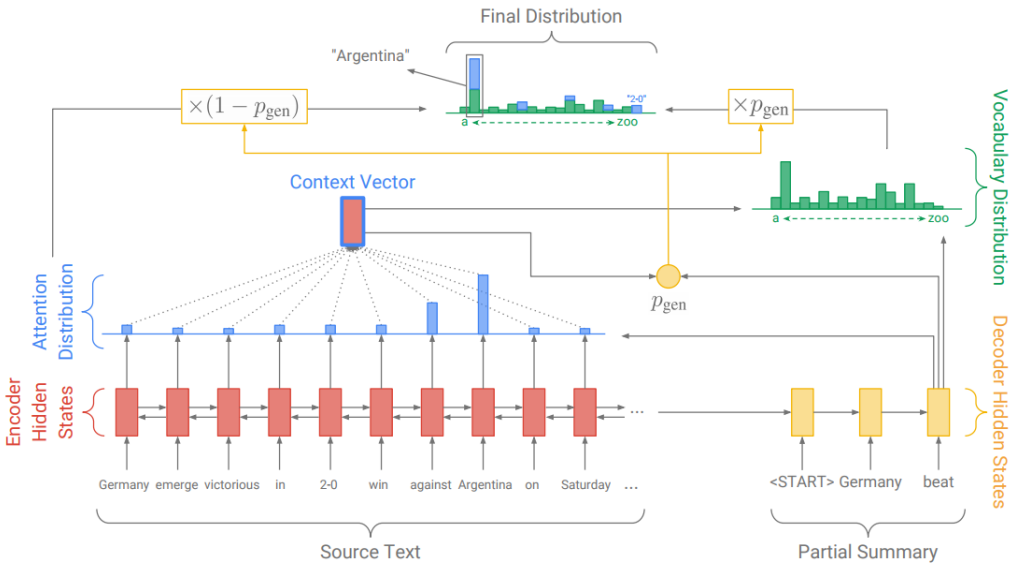

Figure 3.4: Pointer Network as defined by See et al. [36]

Dataset and Training

The dataset used in this paper is the CNN/Daily Mail corpus as was used in Nallapati et al. [24]. For all of their experiments the authors used 128-dimension word embeddings that were learned during training, and hidden states of 256-dimensions. vocabulary size is restricted to 50k words. Maximum size for input article is 400 tokens where the summary is limited by 100 tokens for training and 120 for test time. They trained on a single Tesla K40m GPU with a batch size of 16 using beam search of size 4 during test time. Training took from over 3 to almost 9 days for different models and hyperparameters.

Results

As mentioned above, See et al. [36]'s model ran on the CNN/Daily Mail corpus so the only models that the authors can compare to are abstractive models suggested in Nallapati et al. [24] and extractive models suggested in Nallapati et al. [23]. The suggested models in this paper outperform the best abstractive model suggested by 4 ROUGE points on average, and had competitive results with the extractive model.

3.3.4 A Deep Reinforced Model for Abstractive Summarization

In this paper, Paulus et al. [29] suggest four major components to be added to the attentional encoder-decoder LSTM RNN with pointer network switch: *Intra-Temporal Attention on Input Sequence*, *Intra-Decoder Attention*, *Policy Learning* and *Mixed training objective function*.

Suggested Models

Intra-Temporal Attention on Input Sequence. On top of the basic model described above, the authors suggested to add an intra-temporal attention function in order to reduce repetitions. This kind of attention discourages the model from attending multiple times to the same part of the input. This is done by penalizing the attention weight of input tokens that have already had high attention score in past decoding steps. The weights e'_{ij} are computed using the following method:

$$e'_{ij} = \begin{cases} \exp(e_{ij}) & \text{if } i = 1 \\ \frac{\exp(e_{ij})}{\sum_{k=1}^{i-1} \exp(e_{kj})} & otherwise. \end{cases} \tag{3.8}$$

Where, unlike in Bahdanau et al. [1] attentions. e_{ij} is calculated like the following:

$$e_{ij} = h_i^d W_{attn} h_j^e \tag{3.9}$$

Where h^e, h^d are the encoder and decoder hidden states respectively. All the other parts in the attention mechanism are calculated similarly.

Intra-Decoder Attention. Another method of dealing with repetitions is allowing the decoder to look back at previous decoding steps to enable the model to create more structure predictions. This is done similarly to the way presented above $e_{ii'}$ is calculated only with respect to h^d

Policy Learning. Usually the training paradigm is done with respect to the following "teacher-forcing" loss function:

$$L_{ML} = -\sum_{t=1}^{n'} \log p(y_t^* | y_1^*, \cdots, y_{t-1}^*, x) \tag{3.10}$$

Where y_i^* denotes the ground-truth output token in the i-th decoding step. However minimizing maximum likelihood loss, L_{ML}, doesn't necessarily result with the optimum ROUGE score. There are two possible reasons for this. The first, *exposure bias*, refers to the fact that although the model is fully supervised at train time, during test time, the model doesn't get any indication about its predictions and, therefore, carries errors from previous wrong generations. The second reason is that there are multi-

ple valid methods to summarize a document, but evaluation is done with respect to a single human generated summary.

So instead of minimizing the log maximum-likelihood loss, we can maximize, using reinforcement learning (RL), the metric according to which we evaluate the produced summary.

The authors decided to use the training paradigm called *self-critic* as suggested in Rennie et al. [32] where at each generation step, two tokens will be generated, the first is $\hat{y}$ produced per usual, and the second, y^s that is sampled from the probability distribution $p(y_t^s|y_1^s, \cdots, y_{t-1}^s, x)$. From a RL point of view, $\hat{y}$ is the token maximizing the current language model, by that we are exploiting our model. On the other hand, by sampling from $p(y_t^s|y_1^s, \cdots, y_{t-1}^s, x)$ we are exploring our model.

Given the two tokens, we minimize the following loss function:

$$L_{rl} = (r(\hat{y}) - r(y^s)) \sum_{t=1}^{n'} \log p(y_t^s|y_1^s, \cdots, y_{t-1}^s, x) \tag{3.11}$$

Where $r(y)$ is a rewarding function.

Mixed training objective function The previous model's single purpose is maximizing a specific metric during training, this may cause a decrease of readability of the generated summary. in other words, using the L_{RL} function caused for a lesser language model comparing to the one obtained by using the L_{ML} loss function. For this reason the authors suggested the following loss function:

$$L_{mixed} = \gamma L_{rl} + (1 - \gamma)L \tag{3.12}$$

where γ is a hyperparameter.

Dataset and Training

The datasets used in this paper are the *CNN/Daily Mail* and the *New York Times (NYT)* corpora. The second is a collection of articles published between 1996 and 2007 for the New York Time news agency, summaries in this corpus are usually shorter than in the first.

Like in Nallapati et al. [24], pointer data is added only for OOV words and words that refer either to a person, location, organization, or misc according to Stanford NER [21]. The authors used 200-dimensional LSTMs, and 400-dimensional LSTMs RNN for the encoder and decoder accordingly with 100-dimensional pre-trained GloVe as words embeddings [30]. Vocabulary sizes are 150k and 50k for input and output vocabularies respectively, and as stated before, the $r(y)$ reward function is chosen to be ROUGE-L.

Results

Comparing this paper's suggested models to Nallapati et al. [24]'s model resulted an improvement of between 2 and 6 points for the different ROUGE metrics, with best models being the RL with intra-attention model and the mixed model with intra-attention.

Testing the suggested models on the NYT corpus couldn't be compared to any other model as this paper is the first one to use it for the abstractive summarization task. Although, interestingly enough, results are extremely high compared to the *CNN/Daily Mail* results. This discrepancy might be due to the shorter summaries lengths in *NYT* corpus comparing

to the *CNN/Daily Mail* corpus.

3.4 A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task

In this work we use the model suggested in Chen et al. [3] as our questions answering performance evaluator. Here we will present the model suggested in this paper. A figure illustrating the proposed model is presented in Fig. 3.5.

In this paper the authors suggested a model that is given two inputs, the text and a question and it needs to output the mostly likely answer.

Both the text and question go through an embedding linear layer resulting $p1, \cdots, p_m$ and $q1, \cdots, q_m$ for the text and question respectively. All text embeddings are going through a bi-directional LSTM [15] resulting $\widetilde{p}_i = [\vec{h}_i, \overleftarrow{h}_i]$ where $\vec{h}_i$ and $\overleftarrow{h}_i$ are the different LSTM hidden stats at timestep i. Another LSTM is used to receive q, a fixed length representation of the question.

Then, the authors compare the questions representation q and all the contextual embeddings, and select the pieces of information that are relevant to the question. the authors used a bilinear function producing energy like the following and producing weights and context vector per usual.

$$
\begin{aligned}
o &= \sum_{i=1} a_i \widetilde{p}_i \\
a_i &= softmax_i(e_i) \\
e_i &= q^T W \widetilde{p}_i
\end{aligned}
\tag{3.13}
$$

Prediction is done by taking the most likely answer over all entities that appear in the text in the following way:

$$
a = arg\,max_{a \in p \cap E} W_a^T o \tag{3.14}
$$

3.5 Conclusion

We have reviewed methods to perform abstractive summarization using neural networks. The key ideas that have been developed are:

1. Sequence to sequence / cross-entropy loss [33]
2. Attention [1]
3. Large vocabulary trick [24]
4. Copy mechanism [36, 42]
5. Coverage mechanism [36]
6. Intra-temporal attention [29]

In the following sections we will describe the research objectives for this work and define our method and model to achieve these objectives.

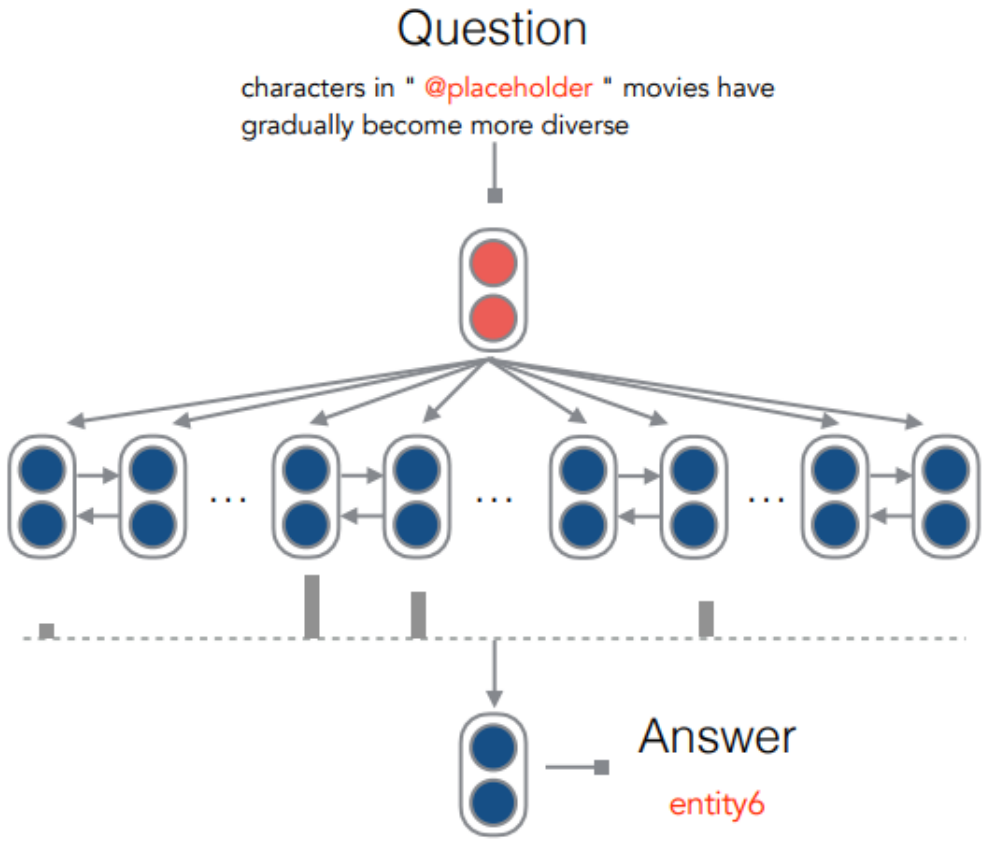

Figure 3.5: Chen et al. [3]'s neural network architecture for the reading comprehension task

4 Research Objectives

We define our research questions addressed in this work:

- **Question #1: Can we define a new evaluation metric designed for single-source datasets? Can we take advantage of the ongoing research in the field of question answering and reduce from the evaluation problem to an question-answering problem?**

  The most popular evaluation metric nowdays, the intrinsic ROUGE metric, known to be problematic for single-source documents [20]. We would like to describe a new evaluation metric that will be useful for this kind of datasets. Additionally, the research field of question-answering has improved immensely in recent times, we believe this will be helpful as a proxy for evaluating summarization models.

- **Question #2: Given a new evaluation metric, can an end-to-end model be designed to maximize this new evaluation metric?**

  While one of our contributions is describing a new evaluation metric, another is suggesting a new training paradigm such that will optimize the suggested metric.

5 APES

5.1 Defining APES

Evaluating a summarization system with APES applies the following method: APES receives a set of news articles summaries, question-and-answer pairs referring to central information from the text and an automatic QA system. Then, APES uses this QA system to determine the total number of questions answered correctly according to the received summaries. We use Chen et al. [3]'s model trained on the *CNN* dataset as our QA system for all our experiments. For a given summarizer and a given dataset, APES reports the average number of questions correctly answered on the basis of the summaries produced by the system.

This method is especially relevant for the main headline generation dataset used in recent years, the *CNN/Daily Mail* dataset, as it was initially created for the question answering task by Hermann et al. [13]. It contains 312,085 articles with relevant questions scraped from the two news agencies' websites. The questions were created by removing different entities from the manually produced highlights to create 1,384,887 fill-in-the-blank questions. The dataset was later repurposed by Cheng and Lapata [4] and

Original Reference Summary:

Arsenal beat Burnley 1-0 in the EPL . a goal from Aaron Ramsey secured all three points . win cuts Chelsea 's EPL lead to four points .

Produces questions:

_____ beat @entity7 1 - 0 in the @entity4
Answer: Arsenal
@entity0 beat _____ 1 - 0 in the @entity4
Answer: Burnley
@entity0 beat @entity7 1 - 0 in the _____
Answer: EPL
a goal from _____ secured all three points
Answer: Aaron Ramsey
win cuts _____'s @entity4 lead to four points
Answer: Chelsea
win cuts @entity19 's _____ lead to four points
Answer: EPL

Figure 5.1: Example 202 from the *CNN/Daily Mail* test set

Nallapati et al. [24] to the summarization task by reconstructing the original highlights from the questions. Fig 5.1 shows an example for creating questions out of a given summary.

While the *CNN/Daily Mail* dataset is a suitable environment for our QA based evaluation metric, similar questions can be created for any news summarization dataset using the same method as in Hermann et al. [13]: given a reference summary, we find all possible entities, (*i.e.*, Name, Nationality, Organization, Geopolitical Entity or Facility) and create fill-in-the-blank type questions where the answers are these entities. We provide code for this procedure and apply it on the DUC/TAC datasets as discussed below.

	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU	APES
Pyramid	0.590	0.468*	0.599	0.563*	**0.608**
Responsiveness	0.540	0.518*	0.537	0.541	**0.576**

Table 5.1: Pearson Correlation of ROUGE and APES against Pyramid and Responsiveness on summary level. Statistically significant differences are marked with *

5.2 APES on the TAC2011 AESOP Task

In order to evaluate if an automatic metric can accurately measure a summarization system performance, it is necessary to measure its correlation to manual metrics. The TAC 2011 *Automatically Evaluating Summaries of Peers* (*AESOP*) task [27] has provided a dataset that includes, alongside the source documents and reference summaries, three manual metrics: Pyramid [26], Overall Responsiveness [8] and Overall Readability. Two sets of documents are provided: the first assumes nothing about the reader of the summary, like in the standard summarization task, while the second set assumes the reader has already read the documents from the first document set (simulating an update summarization task). As our main focus is to assess APES performance on unseen data, we use only the documents from the first set.

In order to evaluate APES on the AESOP dataset, we created the required set of questions as presented in Fig.5.1. We used the same QA system [3] trained on the *CNN* dataset. This system is a competent QA system for this dataset, as both AESOP and *CNN* consist of news articles. Training a QA model on the AESOP dataset would be optimal, but it is

not possible due to the small size of this dataset. Nonetheless, even this incomplete QA system reports valuable results that justify APES value.

While the two datasets are similar, they differ dramatically in the type of topics the articles cover. *CNN/Daily Mail* articles deal with people, or more generally, Named Entities, averaging 6 named entities per summary. In contrast, TAC summaries average 0.87 entities per summary, with very high variance. The TAC dataset is divided into various topics. The first four topics, *Accidents and Natural Disasters*, *Attacks*, *Health and Safety* and *Endangered Resources* average 0.65 named entities per document, making them incomparable to headline generation datasets. The last topic, *Investigations and Trials*, averages 3.35 named entities per document, making it more relevant to simulate a news article dataset. We report correlation below only on this segment of TAC, which contains 204 documents (1/5 of the overall TAC dataset).

We follow the work of Louis and Nenkova [20] and compare input level APES scores with manual Pyramid and Responsiveness scores provided in the AESOP task. Results are in Table 5.1. In *Input level*, correlation is computed for each summary against its manual score. In contrast, *system level* reports the average score for a summarization system over the entire summary dataset. Input level correlation is a good indication of the potential of the metric to guide an automatic system when it optimizes the metric on each given input.

While ROUGE baselines were beaten only by a very small number of suggested metrics in the original AESOP task, we find that APES shows better correlation than the popular R-1, R-2 and R-L and the strong R-SU. While not all hypotheses are statistically significant, APES gives additional

	R-1	R-2	R-L	R-SU	APES
R-1	1.00	0.83	0.92	0.94	0.66
R-2		1.00	0.82	0.90	0.61
R-L			1.00	0.89	0.66
R-SU				1.00	0.67
APES					1.00

Table 5.2: Correlation matrix of ROUGE metrics and APES

value comparing to ROUGE: ROUGE metrics are highly correlated one with another (around 0.9) as shown in Table 5.2, indicating that multiple ROUGE metrics provide little additional information. In contrast, APES is not correlated with ROUGE metrics to the same extent (around 0.6). This suggests that APES provides additional information regarding the text in a manner that ROUGE does not. For this reason we believe APES complements ROUGE.

Louis and Nenkova [20] further show that ROUGE correlation to manual scores tends to drop when reducing the number of reference summaries. While APES is not immune to this, as the number of questions becomes smaller when the number of reference summaries is reduced, it still performs well when reducing the number of references to a single document. In the AESOP dataset, 8 reference summaries are provided by different assessors. When comparing to each assessor separately on Pyramid and Responsiveness, the correlation of APES is highest in 7 out of 16 trials, while that of R1 is highest in 6 trials and RL in 2 trials. In general, correlation between any of the metrics and single references is extremely noisy, indicating that reliance on evaluation on a single reference, which is

standard on large-scale summarization datasets, is far from satisfactory.

We have established that APES achieves equal or improved correlation with manual metrics when compared to ROUGE, and captures different type of information than ROUGE, so that APES can complement ROUGE as an automated evaluation metric. We now turn to assessing recent summarization methods on the APES metric and developing a model that directly attempts to optimize APES.

As pointed out in 7 the headline generation dataset most used in recent years, the *CNN/Daily Mail* dataset [13], was constructed by creating questions about entities from the target reference summary. Since the target summary contains salient information from the source document, we can deduce that all entities appearing in the target summary are *salient entities*.

Reducing summaries evaluation to an extrinsic task such as question answering is intuitively appealing, but this reduction is effective only under specific settings: when questions focusing on central information are available and when an automatic question answering model is reliable enough.

Questions focusing on salient entities can be available as part of the dataset: the headline generation dataset most used in recent years, the *CNN/Daily Mail* dataset [13], was constructed by creating questions about entities that appear in the reference summary. Since the target summary contains salient information from the source document, we consider all entities appearing in the target summary as *salient entities*. In other cases, such as the Gigaword [12] and New York Times [34] datasets, such questions can be generated in an automated manner, as we discuss below.

Given source documents and associated questions, a QA system can be trained over fill-in-the-blank type questions as was shown in Hermann et al. [13] and Chen et al. [3]. Chen et al. [3] achieve 'ceiling performance' for the question answering task on the *CNN/Daily Mail* dataset. We empirically assess in our work whether this performance level (accuracy of 72.4 and 75.8 over CNN and Daily Mail respectively) makes our evaluation scheme feasible and well correlated with manual summary evaluation.

Given the availability of questions and QA models, we propose APES as an evaluation metric for news article datasets, the most popular summarization genre in recent years. We leave expanding this evaluation method for other genres to future work.

News articles include a high number of named entities. When analyzing systems performance on APES (Table 5.3), a system may fail either when it misses a salient entity in the summary, or when it includes the salient entity, but in a context not relevant to the corresponding question. When this happens, the QA system would not be able to identify the entity as an answer to a question referring to the context.

When we compare the average number and type of entities in summaries generated by existing automatic summarizers to that in reference summaries, we observe that the observed models, while producing state-of-the-art ROUGE scores and a high number of named entities (5 vs. 6 on average), fail to focus on salient entities when generating a summary (about 2.6 salient entities are mentioned on average vs. 4.9 in the reference summaries). This observation indicates that optimizing ROUGE does not necessarily lead to better question answering capability. This motivates us to measure APES separately from (and in addition to) ROUGE. Notice

Model	APES	#Entities	#Central Entities
See et al. [36]	38.2	4.90	2.57
Baseline model	39.8	4.99	2.61
Gold Summaries	85.5	6.00	4.90

Table 5.3: Average number of entities and central entities by model

that solely increasing the number of entities is damaging: mentioning too many entities may cause a decrease in the QA accuracy, as it increases the number of possible answers, which would distract the QA system.

A note regarding the validity of single source document. As mentioned, in contrast to the popular single source datasets, AESOP 2011 released a dataset containing 4 reference summaries for each source document. While the single reference dataset is the most common type in recent years, we found worrying signs when we compared ROUGE scores associated with the different assessors. The results, displayed in 5.4 and 5.5 display the correlation between the assessors by their respective ROUGE-1 and ROUGE-2 scores over all TAC 2011 dataset. The low correlation between assessors points to the additional information each summary produced by the assessor introduce to the evaluation metric. This strengthen our original concern that single reference datasets should be optimized with caution.

	A	B	C	D	E	F	G	H
A	1	0.83	0.83	0.81	0.84	0.68	0.7	0.86
B		1	0.74	0.83	0.7	0.73	0.67	0.82
C			1	0.72	0.7	0.79	0.65	0.84
D				1	0.71	0.73	0.81	0.83
E					1	0.72	0.79	0.78
F						1	0.81	0.78
G							1	0.87
H								1

Table 5.4: ROUGE-1 correlation between TAC 2011 assessor

	A	B	C	D	E	F	G	H
A	1	0.61	0.63	0.48	0.67	0.47	0.43	0.68
B		1	0.43	0.64	0.4	0.32	0.29	0.55
C			1	0.45	0.43	0.47	0.61	0.61
D				1	0.43	0.55	0.53	0.62
E					1	0.46	0.51	0.59
F						1	0.67	0.47
G							1	0.6
H								1

Table 5.5: ROUGE-2 correlation between TAC 2011 assessor

6 APESO - APES Optimizer

6.1 Baseline Model

In this section we describe APESO, a model designed to optimize the APES metric. In order to experiment with direct optimization of APES, we reconstruct as a starting point a model which encapsulates the key techniques used in recent abstractive summarization models. Our model is based on the OpenNMT project Gehrmann and Rush [10]. All PyTorch [28] code, including entities attention and beam search refinement is available online[1]. We also include generated summaries and trained models in this repository.

Recent work in the field of abstractive summarization [24, 29, 33, 36] share a common architecture as the foundation for their neural models: an encoder-decoder model [38] with an attention mechanism [1]. Nallapati et al. [24] and See et al. [36] augment this model with a copy mechanism [42]. This architecture minimizes the following loss function:

[1] `https://github.com/mataney/OpenNMT-py-EMNLP`

$$\text{loss}_t = -\log P(w_t^*) \tag{6.1}$$

$$\text{loss} = \frac{1}{T_y} \sum_{t=0}^{T_y} \text{loss}_t \tag{6.2}$$

$loss_t$, is the negative log likelihood of generating the gold target word w_t^* at timestep t where $P(\cdot)$ is the probability distribution over the vocabulary (or the *extended vocabulary* for the augmented copy mechanism model). The total loss of generating a sequence is the average over all the losses at each timestep t. We refer the reader to See et al. [36] for a more detailed description of this architecture.

Unlike See et al. [36], we do not train a specific coverage mechanism to avoid repetitions. Instead, we incorporate Wu et al. [43]'s refinements of beam search in order to manipulate both the summaries' coverage and their length. In the standard beam search, we search for a sequence Y that maximizes a score function $s(Y, X) = \log(P(Y|X))$. Wu et al. [43] introduce two additional regularization factors, *coverage penalty* and *length penalty*. These two penalties, with an additional refinement suggested in **?**], yield the following score function:

$$\begin{aligned} s(Y, X) &= \log(P(Y|X))/lp(Y) - cp(X; Y) \\ lp(Y) &= \frac{(5 + |Y|)^\alpha}{(5 + 1)^\alpha} \\ cp(X; Y) &= \beta(-T_X + \sum_{i=1}^{T_X} \max(\sum_{j=1}^{T_Y} a_{i,j}, 1.0)) \end{aligned} \tag{6.3}$$

where α, β are hyper-parameters that control the length and coverage penal-

ties respectively and $a_{i,j}$ is the attention probability of the j-th target word on the i-th source word.

$cp(X;Y)$, the coverage penalty, is designed to discourage repeated attention to the same source word and favor summaries that cover more of the source document with respect to the attention distribution.

$lp(Y)$, the length normalization, is designed to accurately compare between beam hypotheses of different length. In general, beam search favors shorter outputs as log-probability is added at each step, yielding lower scores for longer sequences. lp compensates for this tendency.

In the following section, we describe how we extend this baseline model in order to maximize the APES metric. The new model learns to incorporate more of the salient entities from the source document in order to optimize its APES metric.

6.2 Entities Attention Layer

As we observed before, failure to capture salient entities in summaries is one cause for low APES score. In order to drive our model towards the identification and mention of salient entities from the source document, we introduce an additional attention layer that learns which are the important entities of the source document. We hypothesize that these entities are more likely to appear in the target summary, and thus are better candidate answers to one of the salient questions for this document.

We learn for each word in the source document its probability of belonging to a salient entity mention. We adapt the classical soft attention

Model	APES	ROUGE		
		1	2	L
Source	61.1	-	-	-
Gold-Summaries	85.5	100	100	100
Shuffled Gold-Summaries	30.9	100	7.0	58.3
Lead 3	45.1	40.1	17.3	36.3
Pointer-generator + coverage [36]*	38.2	39.3	16.9	35.7
Baseline model	39.8	39.3	17.3	36.3
Our model	**46.1**	**40.2**	**17.7**	**37.0**
Our model with gold entities positions	46.3	40.4	17.8	37.3

Table 6.1: APES: Percent of questions answered correctly using by document. *Obtained from the model uploaded to `github.com/abisee/pointer-generator`

mechanism of Bahdanau et al. [1]: after encoding the source document, we run an additional single alignment model with an empty query and a sigmoid layer instead of the standard softmax layer.

$$\begin{aligned} a_j^e &= \sigma(e_j^e) \\ e_j^e &= v^T \tanh(Uh_j + b) \end{aligned} \tag{6.4}$$

where U, b, v are learnable weight matrices, h_j is the encoder hidden state for the j-th word and $\sigma(\cdot)$ is a logistic sigmoid function. a_j^e reflects the probability of the j-th token of being a salient entity.

We used Yang et al. [44]'s suggestion of using an empty attention query, as the alignment we learn is independent of the different decoding steps.

The second modification in this attention layer with respect to Bah-

danau et al. [1] is that we replace the softmax function with a sigmoid: while in the standard alignment model, we intend to obtain a normalized probability distribution over all the tokens of the source document, here we would like to get a probability of each token being a salient entity independently of other tokens.

In order to drive this attention layer towards salient entities, we define an additional term in the loss function.

$$loss_e = BCE(a^e, s^*) \tag{6.5}$$

where s^* is a binary vector of source length size, where $s^*_j = 1$ if x_j is a salient entity, and 0 otherwise, and BCE is the binary cross entropy function. This term is added to the standard log-likelihood loss, changing equation (6.2) to the following composite loss function:

$$\text{loss} = \delta\ loss_e + (1 - \delta)\frac{1}{T_y}\sum_{t=0}^{T_y}\text{loss}_t \tag{6.6}$$

where δ is a hyper-parameter. We join these two terms in the loss function in order to learn the entities attention layer while keeping the summarization ability learned by Eq. (6.2).

@@ Add diagram of the architecture

6.3 Entities Attention and Beam Search

After the attention layer has learned the probability of each source token to belong to a salient entity, we pass the predicted alignment to the beam search component at test-time. Using this alignment data, we wish to encourage beam search to favor hypotheses attending salient entities.

Accordingly, we introduce a new term ep to the beam search score function of equation (6.3):

$$\begin{aligned} s(Y,X) &= \log(P(Y|X))/lp(Y) - cp(X;Y) \\ &\quad - ep(X;Y) \\ ep(X;Y) &= \gamma \sum_{i=1}^{T_X} \max(a_i^e - \sum_{j=1}^{T_Y} a_{i,j}, 0.0) \end{aligned} \tag{6.7}$$

$ep(X;Y)$ penalizes summaries that do not attend parts of the source document we believe are central.

Fig.9.1 compares summaries produced by this model and the baseline model by showing their respective attention distribution and the impact on the decision of which words to include in the summary based on the attention level derived from salient entities.

7 Dataset

The *CNN/Daily Mail* dataset, originally created by Hermann et al. [13] for the passage-based reading comprehension task, consists of articles from the CNN and Daily Mail websites. The dataset includes the article body, and a collection of questions produced from human generated bullets. For every bullet, a different entity was removed to get fill-in-the-blank type questions, resulting in a total of 312,085 documents and 1,384,887 questions in the dataset. This task can be seen as a multiple-choice questionnaire since the number of possible answers is determined by the number of entities in the input document. On average, the number of entities in each document is 23.02, making the task of factoid question answering on this corpus less challenging than the standard question-answering task such as SQuAD [31]. We will note that for a small number of documents, the correct answer does not appear in the source document and only appears in the bullets.

Previous automatic summarization works [4, 24] suggested to re-purpose the constructed questions for the summarization task while using the body of the article as the source document and concatenate the bullets as the target document. The resulted dataset consists of the 312,085 source-target pairs, the same number of documents as in the Reading-Comprehension

corpus, where the average source document size is 766 words with 29.74 sentences while summaries are 53 words and 3.72 sentences on average.

In the original paper [13], the authors suggested masking the entities in the text with a special *entityX* token, representing the specific word is an entity with ID *X*. One advantage for this representation is that multi-word entities can be represented as a single token. While this representation is helpful in the technical process of picking an entity as the candidate answer, this representation might miss certain attributes regarding the entity. *e.g.*, POTUS is a president of a country called United States. While a language model might learn such dependencies, using the *entityX* representation can not. Nallapati et al. [24], being the first paper to use this dataset for the summarization task, used the representation suggested by Hermann et al. [13]. See et al. [36] are the first to include the entities as originally displayed in the news agencies without the special *entityX* token.

While both of these datasets are available online, a mapping from the article / summary pair to its respective questions was unavailable until now. We make this mapping available online to enable reproduction of this evaluation[1] together with all the source code of our experiments.

[1] `https://github.com/mataney/Anonymized-CNN-Daily-News-Stories`

8 Experimental Settings

For our experiments, we used a Bidirectional LSTM encoder with 256-dimensional hidden states for each direction, an LSTM decoder with 512-dimensional hidden states and 128-dimensional embeddings for a 50k shared-vocabulary words. We do not use pretrained word embeddings.

We use the Adagrad [9] optimizer with a starting learning rate of 0.15 and gradient clipping with a maximum gradient norm of 2. At train time source and target documents are truncated to 400 and 100 tokens respectively. After training our baseline model for 20 epochs, we fine-tune the network with equation (6.6) loss for an additional 5 epochs starting again with 0.15 as initial learning rate. Results reported in this work correspond to $\delta = 0.01$.

At test-time, we do not truncate the source documents enabling the network to attend over all input text. We use Eq. (6.7) as the beam search score function, penalizing using $cp(X;Y)$ every single decoding step and $lp(Y)$ and $ep(X;Y)$ only when all hypotheses are done. We choose α, β, γ values of $0.9, 0.5, 0.5$ respectively for our model. We also used Paulus et al. [29] suggestion of repetition avoidance by blocking trigrams appearing more than once at inference time.

We used grid search for hyperparameters optimization. Given Wu et al. [43] and Gehrmann and Rush [10] suggestions for α, β values we ran a grid search over $\alpha \in [0.2, 0.9, 1.5], \beta \in [0.5, 1, 2, 5, 10]$ and also $\gamma \in [0.5, 1, 2, 5, 10]$. α, β, γ values of $0.9, 0.5, 0.5$ yields best ROUGE and APES scores combination over the validation set (APES, R1, R2, RL scores of 0.467, 0.412, 0.184,0.381 respectively)

Running APES evaluation on a generated test set (of size 11,490 summaries) takes about 40 minutes using a single process.

9 Results

We report our results in Table 6.1. For each system, we present its APES score alongside its F1 scores for ROUGE-1, ROUGE-2 and ROUGE-L, computed using pyrouge. [1].

We first report APES results on full source documents and gold summaries, in order to assess the capabilities of the QA system used for APES. A simple answers extractor could answer 100% of the questions given the gold-summaries. But the QA system is trained over the source documents, and learns to generalize and not "just" extract the answer. While gold-summaries present a very high APES score, the score reported for the source documents (61.1%) is a realistic upper bound for APES.

We then present shuffled gold-summaries, where we randomly shuffled the location of each unigram in the gold summary. This score shows that even when all salient entities are in the shuffled text, APES is sensitive to the loss of coherence, readability and meaning. This confirms that APES does not only match the presence of entities. In contrast, ROUGE fails to punish such incoherent sequences. Finally, we report ROUGE and APES for the strong Lead 3 sentences of the source document - a baseline known

[1] `https://pypi.org/project/pyrouge/`

to beat most existing abstractive methods.

We then present APES and ROUGE scores for abstractive models, See et al. [36]'s model, our baseline model and our APES-optimized model. Our model achieves significantly higher scores for APES (46.1 vs. 39.8) and slightly improves all ROUGE metrics (by about 1 F-point over the baselines). It is interesting that while our objective is maximizing APES score, our model also increases its ROUGE scores. Unlike Paulus et al. [29] where the authors suggested a RL based model to specifically optimize ROUGE, we optimize a different metric and receive competitive ROUGE results.

We finally report the results obtained by our model when gold salient entities positions are given as oracle inputs instead of the predicted a^e scores. The corresponding score (46.3 vs. 46.1) is only slightly above the score obtained by our model. This indicates that the component of our model predicting entity saliency is good enough to drive summarization.

We carried out an informal error analysis to examine why some summaries perform worse than others with our architecture. We compared summaries that produce perfect APES score (1,630 out of 11,490 total) to the summaries with zero APES score (1,691). We measure the density of salient named entities in the source document: #(salient entity mentions)/#(distinct salient entities). This density in the case of perfect APES summaries is much higher than that for low APES summaries (4.9 vs. 3.6). This observation suggests that we fail to produce higher APES scores when the salient entities aren't marked through sheer repetition.

9.1 Hypothesis correctness

- **Hypothesis #1: Can we define a new evaluation metric designed for single-source datasets? Can we take advantage of the ongoing research in the field of question answering and reduce our evaluation problem to an question-answering problem?**

  We define APES as our novel evaluation metric, APES is indeed using a pretrained Question-Answering system and by that taking advantage of the breakthroughs in this field. We also managed to show its value by correlating it to the TAC 2011 dataset. While we believed this correlation can be generalized to any type of source document with respective questions, we only managed to show APES superiority over ROUGE on a specific type of source document.

  APES encounters low correlation with manual evaluation methods, and by that might limit its value. This happens for documents with low density of named entities, for example, articles regarding *Accidents and Natural Disasters*, *Attacks*, *Health and Safety* and *Endangered Resources* from the AESOP 2011 dataset. This stories are not about named entities, this result our lack of correlation to manual metrics. For articles with low-density of entities, we believe creating questions, using methods researched in the field of Question-Generation, will be helpful.

- **Hypothesis #2: Given a new evaluation metric, can an end-to-end model be created designed to maximize this new evaluation metric?**

  We managed to build an end-to-end neural model optimizing the

APES score, and by that increasing ROUGE scores comparing to the base model.

The novelty in our model assumes a sufficient number of entities, of whom the article surround. As pointed before, other types of articles, different from articles in *CNN/Daily Mail* , might require an additional change in our proposed model.

Additionally, from a QA point of view, the QA system trained for this task suffers from co-reference detection. a QA system that can overcome this problem will be beneficial to our cause and emphasis APES' value over intrinsic evaluation methods like ROUGE.

Source document:

jack wilshere may rub shoulders with the likes of alexis sanchez and mesut ozil on a daily basis but he was left starstruck on thursday evening when he met brazil legend pele . even better for wilshere , the arsenal midfielder was given the opportunity to interview the three-time world cup winner during the launch party of 10ten talent . both wilshere and pele , along with glenn hoddle , are clients and the england international made sure his fans on twitter knew about their meeting by posting several tweets . brazil legend pele -lrb- left -rrb- and arsenal midfielder jack wilshere pose for a photo during launch of 10ten talent . wilshere was given the ' honour to interview the legendary pele and asked twitter questions from fans . earlier on thursday , wilshere tweeted : ' looking forward to meeting @pele tonight . i ll be asking the best questions you sent . #jackmeetspele . the 23-year-old then followed this up with several tweets about the event , many of which included photos of pele . meanwhile , pele has acknowledged that last year s world cup was a ' disaster for brazil but is not surprised how quickly the likes of oscar and ramires have bounced back in the barclays premier league this season . brazil were humiliated by germany in a 43107 semi-final defeat and the hosts were then thrashed 3-0 by holland in the third-place play-off . pele scored 77 goals in 92 games for brazil and won the world cup three times but the former santos striker still finds last year s capitulation difficult to understand .

Target Summary:

jack wilshere was joined by former **england** manager **glenn hoddle**. the **arsenal** midfielder interviewed **pele** at launch of **10ten talent**. **pele** scored 77 goals in 92 games for **brazil** and won three world cups. the **brazil** legend says the 2014 world cup performance was not expected. the hosts were humiliated 7-1 by **germany** in the semi-finals last summer. **pele** is, however, not surprised by reaction of **oscar** and **ramires** this year.

Baseline Model Prediction:

jack wilshere was given the opportunity to interview the three-time world cup winner. both wilshere and pele are clients and the england international. pele has acknowledged that last year's world cup was a 'disaster'

Our Model Prediction:

jack wilshere was given the 'honour to interview the legendary pele' and asked twitter questions from fans. pele has acknowledged that last year's world cup was a 'disaster' for brazil but is not surprised how quickly the likes of oscar and ramires have bounced back in the premier league this season. the brazil legend scored 77 goals in 92 games for brazil and won the world cup three times.

Figure 9.1: Example 4134 from the *CNN/Daily Mail* test set. Colors in the source reflect differences between baseline and our model attentions: Red reflects words attended by baseline model and not our model, Green reflects the opposite. Entities in **bold** in the target summary are answers to the example questions.

10 Error Analysis

As mentioned before our main contributions are the salient entity attention layer and the change in the beam search that resulted better APES score. This method has some side effects as well, one of them is that beam search is favoring longer summaries as it raises probability of producing more salient entities. These longer summaries are sometimes manifested with sudden endings. Another side effect is that our model is encouraged to produce sentences with a long list of entities.

These problems are displayed in the next examples:

Reference Summary:

raheem sterling has rejected a new contract worth # 100,000-a-week . liverpool boss brendan rodgers says sterling wo n't be sold this summer . his achievements at 20 are not on the same level as some anfield greats . steven gerrard won three trophies in the first year out of his teens . michael owen had scored more than 50 career goals for liverpool . sterling has a long way to go if he is to fulfil his huge potential .

Our model's Generated summary:

michael owen had been named pfa young player of the year . there were also 18

england caps and he had passed 50 career goals for liverpool . steve mcmanaman ended his first season as a liverpool regular with a man-of-the-match performance in the 1992 fa cup final win over sunderland . steven gerrard had won three trophies in the first year out of his teens saw him collect the league cup , the fa cup and the uefa cup . robbie fowler

The last sentence is stopped abruptly as it did not generate full stop suggesting beam search favored beam with high entities attention distribution than a high language model probability.

Another similar summary prediction:

Reference Summary:

qatar 's supreme committee unveil new 40,000-seat stadium for world cup . the al rayyan stadium is fifth 2022 venue and will be completed in 2019 . stadium features ' cooling technology ' for fans ' and players ' comfort .

Our model's Generated summary:

qatar have unveiled the design for their fifth world cup stadium , a 40,000-seater ground built in ' the gateway to the desert ' the al rayyan stadium is inspired by sand dunes and its design has been produced by uk-based architecture firms ramboll and pattern . fans in the stands will sit in a comfortable 24-28 degrees celsius , while paths leading to the stadium will also be cooled . al rayyan will hold matches up to the quarter-final stage of the world cup in qatar in 2022 .

This is a similar problem resulted by our models preference of longer sequences. Notice the relatively short reference summary comparing to the long, entities frequent, summary our model generated.

Reference Summary:

the fbi cites social media messages sent by keonna thomas , 30 . she 's accused of trying to travel overseas to join isis . thomas is one of three women facing federal terror charges this week .

Our model's Generated summary:

keonna thomas , 30 , also known as " young lioness " and " fatayat al khilafah " on march 26 , thomas purchased a ticket to barcelona , with a march 29 departure and an april 15 return to the united states . in the past 18 months , noelle velentzas , 28 , and her former roommate , asia siddiqui , were arrested in new york .

This is an example if a generated summary our models produced that favored long sentences with multiple entities. While this indeed increases the number of entities in our model it does not increase APES score necessarily.

11 Conclusion

In this work we introduced APES, a new automatic summarization evaluation metric for news articles datasets based on the ability of a summary to answer questions regarding salient information from the text. This approach is effective in domains that focus on named entities - such as news articles, where named entities are effectively aligned with Pyramid SCUs. In other non-news domains, other methods for generating questions should be designed. We compare APES to manual evaluation metric on the TAC 2011 AESOP task and confirm its value as a complement to ROUGE.

We introduce a new abstractive model that optimizes APES scores on the *CNN/Daily Mail* dataset by attending salient entities from the input document, which also provides competitive ROUGE scores.

Our major contributions in this work are two fold: Defining and showing APES correlation to manual scores, and presenting a new abstractive model that maximizes APES by increasing attention to salient entities, while increasing ROUGE to competitive level.

Our model optimizes APES performance while preserving competent ROUGE scores by introducing a new attention layer to identify salient

entities and driving the algorithm at search time through a carefully designed score function in the Beam-Search component. While beam search was initially created in order to avoid test-time search errors that a greedy algorithm might produce, a recent paper by Goyal et al. [11] might be helpful for our work as it enables contiguous relaxation for beam search at train-time.

Bibliography

[1] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[2] Celikyilmaz, A., Bosselut, A., He, X., and Choi, Y. (2018). Deep communicating agents for abstractive summarization. *arXiv preprint arXiv:1803.10357*.

[3] Chen, D., Bolton, J., and Manning, C. D. (2016). A thorough examination of the cnn/daily mail reading comprehension task. *arXiv preprint arXiv:1606.02858*.

[4] Cheng, J. and Lapata, M. (2016). Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252*.

[5] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

[6] Chopra, S., Auli, M., Rush, A. M., and Harvard, S. (2016). Abstractive

sentence summarization with attentive recurrent neural networks. In *HLT-NAACL*, pages 93–98.

[7] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

[8] Dang, H. T. (2005). Overview of duc 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12.

[9] Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

[10] Gehrmann, S. and Rush, A. M. (2017). Opennmt: Open-source toolkit for neural machine translation. In *Proc. ACL*.

[11] Goyal, K., Neubig, G., Dyer, C., and Berg-Kirkpatrick, T. (2017). A continuous relaxation of beam search for end-to-end training of neural sequence models. *arXiv preprint arXiv:1708.00111*.

[12] Graff, D., Kong, J., Chen, K., and Maeda, K. (2003). English gigaword. *Linguistic Data Consortium, Philadelphia*, 4:1.

[13] Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

[14] Hobson, S. P., Dorr, B. J., Monz, C., and Schwartz, R. (2007). Task-based evaluation of text summarization using relevance prediction. *Information Processing & Management*, 43(6):1482–1499.

[15] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[16] Hovy, E., Lin, C.-Y., Zhou, L., and Fukumoto, J. Automated summarization evaluation with basic elements. Citeseer.

[17] Jean, S., Cho, K., Memisevic, R., and Bengio, Y. (2014). On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007*.

[18] Jing, H., Barzilay, R., McKeown, K., and Elhadad, M. (1998). Summarization evaluation methods: Experiments and analysis. In *AAAI symposium on intelligent summarization*, pages 51–59.

[19] Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain.

[20] Louis, A. and Nenkova, A. (2013). Automatically assessing machine summary content without a gold standard. *Computational Linguistics*, 39(2):267–300.

[21] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60.

[22] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

[23] Nallapati, R., Zhai, F., and Zhou, B. (2017). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. *hiP (yi= 1— hi, si, d)*, 1:1.

[24] Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., et al. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

[25] Narayan, S., Cohen, S. B., and Lapata, M. (2018). Ranking sentences for extractive summarization with reinforcement learning. *arXiv preprint arXiv:1802.08636*.

[26] Nenkova, A., Passonneau, R., and McKeown, K. (2007). The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(2):4.

[27] Owczarzak, K. and Dang, H. T. (2011). Overview of the tac 2011 summarization track: Guided task and aesop task. In *Proceedings of the Text Analysis Conference (TAC 2011), Gaithersburg, Maryland, USA, November*.

[28] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

[29] Paulus, R., Xiong, C., and Socher, R. (2017). A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

[30] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

[31] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

[32] Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. (2017). Self-critical sequence training for image captioning. In *CVPR*, volume 1, page 3.

[33] Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

[34] Sandhaus, E. (2008). The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

[35] Sankaran, B., Mi, H., Al-Onaizan, Y., and Ittycheriah, A. (2016). Temporal attention model for neural machine translation. *arXiv preprint arXiv:1608.02927*.

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

[37] Steinberger, J. and Ježek, K. (2012). Evaluation measures for text summarization. *Computing and Informatics*, 28(2):251–275.

[38] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

[39] Tay, Y., Phan, M. C., Tuan, L. A., and Hui, S. C. (2017). Skipflow:

Incorporating neural coherence features for end-to-end automatic text scoring. *arXiv preprint arXiv:1711.04981*.

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

[41] Vadlapudi, R. and Katragadda, R. (2010). On automated evaluation of readability of summaries: Capturing grammaticality, focus, structure and coherence. In *Proceedings of the NAACL HLT 2010 student research workshop*, pages 7–12. Association for Computational Linguistics.

[42] Vinyals, O., Fortunato, M., and Jaitly, N. (2015). Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.

[43] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

[44] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.