Neural Contextual News Summarization

Matthias Nitsche
Department of Computer Science
Hamburg University of Applied Sciences
Hamburg, Germany
matthias.nitsche@haw-hamburg.de

Abstract

Multi-document abstractive summarization poses a challenge in *NLP* and deep learning research. On real world data with changing and differing vocabulary such as the news domain it is inherently hard to extract the useful parts of a text. Recent developments have shown that transfer learning is a suitable way to initialize inputs with pre-trained language models. Additional information can be leveraged using attention and preprocessing strategies such as hierarchical encoders and relation graphs. We will put extensive focus on the initialization, intermediate states and knowledge representations, instead of building complex neural network architectures. In this paper we give an overview of the key challenges and state-of-the art concepts for multi-document summarization systems and how they can be applied to real world data.

Keywords - Natural Language Processing (NLP), Multi-Document Summarization, Context, Embeddings, Language Models, Transfer Learning

1 Research Question

The task of multi-document summarization is shortening large amounts of documents keeping their original points and presenting them in an extractive and abstractive fashion. Due to larger and more complex data collection systems and the immense quantity of daily news, the need for grouping and shortening information becomes increasingly useful. While extractive summarization systems focus on reusing, rearranging and dropping words/sentences, abstractive summarization is focused on generating new sentences. Major questions to ask in such systems are:

1. What is useful information to keep? What are the major points?
2. Is the newly generated summarization grammatically and semantically sound?
3. Is the summarization true to the facts presented in the original?
4. What is the general topic connecting multiple documents?
5. Is there a discourse? Are there different or contradicting arguments?

Substantial research and progress has been made in this area, however deep learning is still a new modelling technique applied to summarization. Natural language processing and deep learning are constantly evolving fields. While classical approaches are largely based on finding methods to extract useful features from unstructured text, deep learning models extract features without much consideration. In recent years there is an increasing concentration on inductive biases and prior initialization like embeddings. The research question of this paper is as follows: How to create an abstractive multi-document summarization system that leverages context and side information?

A rough sketch of such as system can be seen in figure 1. The representation ranges from simple statistical count based formats to probabilistic models such as embeddings and language models.

Since side information and contextual information are important aspects to consider when dealing with unstructured text, our focus is on the initialization and representation of text based models.

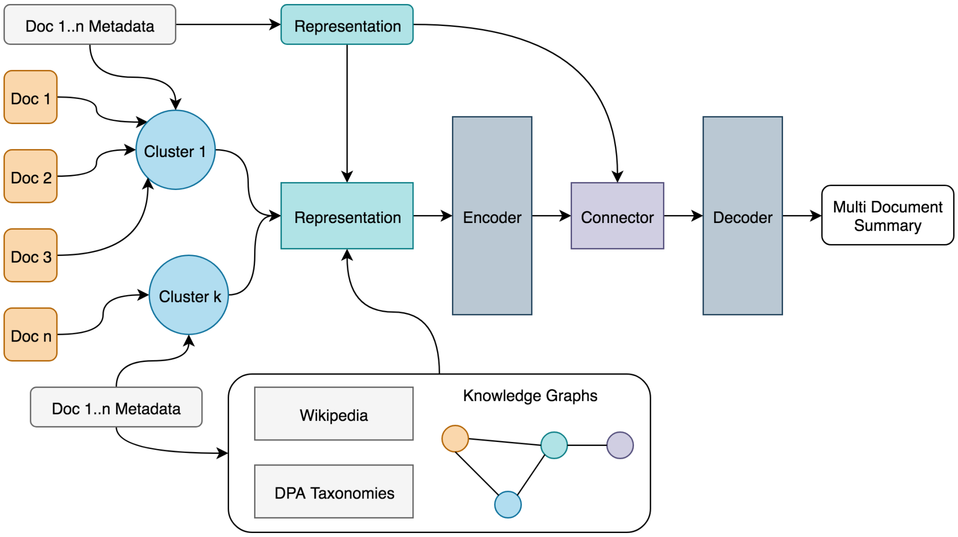


Figure 1: Multi document summarization system

For five years the status quo of *NLP* with deep learning was to initialize models with word embeddings Mikolov et al. (2013); Pennington et al. (2014) yielding 2-3% points of additional accuracy. Recent discoveries by Peters et al. (2018); Howard and Ruder (2018); Radford (2018) have shown that using language models conditioned on the task $P(w_{i+1}|w_i)$, predicting the next word w_{i+1}, generalizes to almost all *NLP* tasks, when trained correctly. These models capture features such as structure, meaning and semantics hierarchically. Behind this idea is the concept of transfer learning, specifically domain adaptation, claiming that it is possible to extract features from your target domain data via a general language model, learned on a different source domain, that is often much larger. The language model is then used to jointly learn a specific task, like summarization. The novelty is, in comparison to word embeddings, that hierarchical features are fine tuned given the specific domain for a specific task. We will investigate how language models and word embeddings can be used for summarization.

When working with real world data sets that were not specifically collected for a certain task, the underlying assumptions of domain adaption are harder to execute. A word only makes sense in combination of its surrounding words, time, social context and factual knowledge. If the collected text does not represent its domain well, is badly written and labels are inadequately set, the best model cannot deal with the deficiencies. We will be working with a real world dataset from the Deutsche Presse Agentur (DPA).

1.1 News and Real Data

The news domain has some interesting properties and often relevant metadata. The DPA dataset has a publishing date, short description, lists of referenced articles and hyperlinks, named entity descriptions, geographical locations, keywords and curated categories as well as topological hierarchies. The articles are of higher quality since it is not a tabloid newspaper with low quality content and contains on average 550 words per article.

The main goal is to use the full text and train it on the descriptions and keywords to learn short descriptive summaries at first. Since the deep learning models alone do not yield any kind of useful or interesting results, we will experiment with different word embeddings and language models. The models are trained on *Wikipedia*, as well as the described metadata. Side information, as we will later see, can be embedded via attention during training.

Since there is a high amount of topological and keyword metadata, it is possible to use different co-reference resolution and clustering algorithms to find articles that should be summarized in groups. It is also possible to use these hierarchies directly and assume their relationship through their respective keywords. For an extensive overview and introduction to clustering algorithms see Aggarwal and

Reddy (2013). This gives the summarizer the competitive advantage of coherent groupings and a lower noise ratio due to outliers.

1.2 Methodology

The methodology is based on the Knowledge Discovery in Databases (KDD) as well as Box's Loop Blei (2014), which is an iterative approach that compromises several steps like data storage, feature extraction, information retrieval combined with a target task. Box's Loop as seen in figure 2 boils down to creating a model (hypothesis), making inference on the data (including their initialization), criticizing the results and improving the model/inference if necessary.

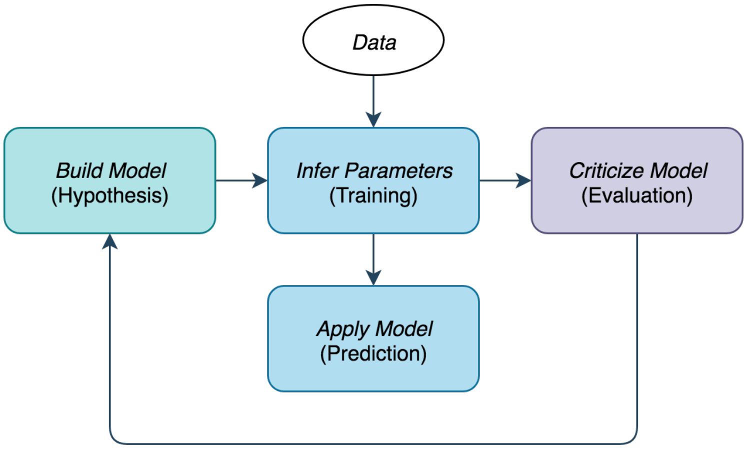


Figure 2: Box loop Blei (2014)

Deep learning is a rather new sub field of machine learning (with substantial research interest), changes are coming by the hour and it is often unclear how to reproduce the training step, given the amount of tunable parameters. Box's Loop in theory enforces a strict testing and book keeping regime. It predicates the hypothesis first, the model and inference as the second step.

2 Theory

The theoretical aspects of document summarization come from classical approaches that leverage well made heuristics and representations of the data. In the following sections we will glance over unsupervised learning, language models, topic modelling and language generation. Abstractive document summarization is an unsolved task with a lot of ideas. We will first give an overview of document summarization, then multi-document summarization and finally language models and embeddings.

2.1 Document Summarization

Document summarization, as explained before, is shortening a text to the relevant points pertaining grammatical and semantical correctness. Allahyari et al. (2017) categorize summarization into either extractive or abstractive. Classical approaches are based on three premises a.) cue words from cue dictionaries pointing to relevant passages, b.) using the title pointing to relevant passages or c.) locations, e.g. most information is in the first to third sentence.

Since automatic document summarization is relatively new to the neural architecture domain, the common way of summarizing is clustering and frequency based approaches such as *tf-idf*. Often this is combined with topic modelling like Latent Semantic Analysis (*LSA*) by Deerwester et al. (1990) or Bayesian topic models like Latent Dirichlet Allocation (*LDA*) by Blei et al. (2003). These algorithms reduce the word dimensionality into fixed size topic distributions pointing to their relevant words. Words grouped under a certain topic distribution are then matched with source sentences that in turn are grouped into certain topics. Summarization is then just the top n matching sentences given their most important words. Since abstractive summarization needs a generator that has sense on how to generate text is not a real topic in the classical domain. Most algorithms deal with abstraction and therefore copying. See for example Gong and Liu (2001).

More recent deep learning techniques use sequence encoder-decoder with attention directly on unstructured text, given a target summary Rush et al. (2015) as gold standard. The encoder learns a representation of the input text, the attention mechanism connects the most useful ideas of the encoder and feeds the output into the decoder which jointly learns a representation given the summary. Later on it is then possible to use the encoder-decoder model to generate new abstractive one sentence summaries with beam search. The limitations are fixed length summaries, duplicate sentences and small documents only.

Nallapati et al. (2016) improve on several points, changing the encoder-decoder objective to a machine translation architecture. The inputs are concatenated to a feature rich encoder using words, POS embeddings, TF and IDF embeddings - represented as discrete one hot bins - as well as NER embeddings. The encoder chains an additional layer capturing hierarchical structures from word to sentence level. *OOV* words are handled via a pointer switch that copies passages that cannot be abstractively generated (due to missing words). They managed to outperform before going neural abstractive systems and introduce the novel pointer mechanism which heavily influences later work.

Pointer generator networks by See et al. (2017) improve on the initial idea by Nallapati et al. (2016), generating or copying depending on what information is available. As can be seen in figure 3, See et al. (2017) use an additional attention mechanism in combination with a coverage vector, eliminating duplicate information, achieving state-of-the-art results.

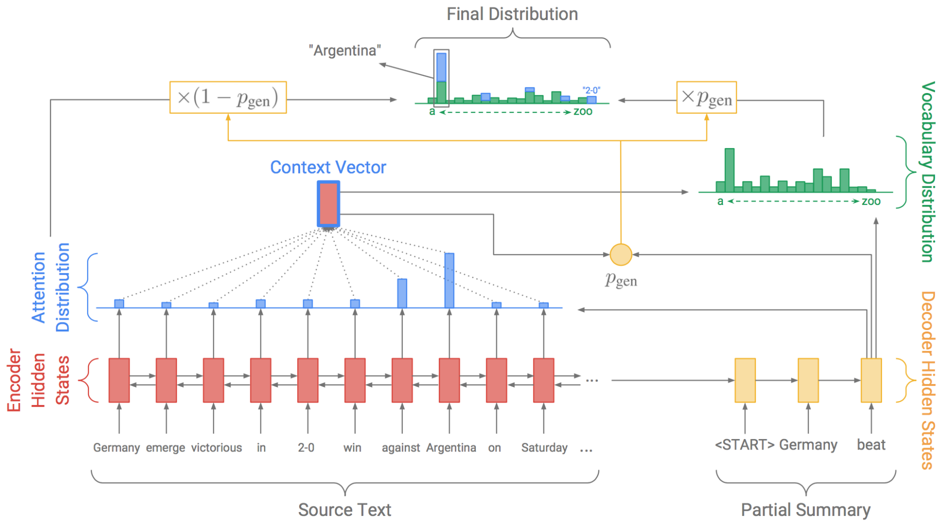


Figure 3: Document summarization using pointer generator networks See et al. (2017)

This architecture solves the problem with fixed length summaries and works with longer sequences. Through the pointer mechanism they can effectively copy passages where *OOV* words occur. The context vector is the result of attending to the input sequence. This context vector is essentially a vocabulary distribution. A generator calculates the probability of whether copying or generating is the best strategy. Due to the coverage vector and improved pointer mechanism the input length can be increased without stepping into catastrophic forgetting or vanishing gradient problems.

The most present problem in document summarization is often generating fluent and readable content as well as long documents. In cases of multi-document summarization the sources can be extremely long. Paulus et al. (2017) found a way to successfully use reinforcement (teacher) learning during training. Instead of using a simple attention mechanism that is applied on the entire output of the encoder, they attend temporarily and found that longer documents got more fluent. This solves the common problems of catastrophic forgetting or vanishing gradient, claiming that through a global reinforcement learning objective and local summary alignment the model is more robust.

This introduces an exposure bias since local summary alignment is not optimal within the global perspective.

In times where there is plenty of news with little information, low entropy, factual checking and cross validation is necessary. This inherent problem is present even when considering the original documents to be factually plausible. Cao et al. (2017) have shown that nearly 30% of the outputs from state-of-the-art models suffer from factual inconsistencies. They reduced this number by 80% yielding an overall 24% improvement. This is done with a standard encoder-decoder architecture and a relation encoder that parses facts from sentences via dependency trees. As can be seen in figure 4 these relations are chained with a dual attention mechanism, one for sentences and one for the dependency tree. Factual parsing via dependency trees is a long standing problem and while not fully solved, classical parsing techniques improve the factual accuracy by large margins.

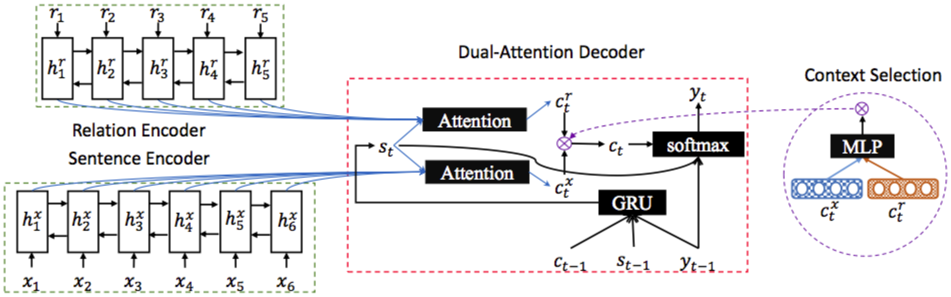

r_1 r_2 r_3 r_4 r_5
h_1^r h_2^r h_3^r h_4^r h_5^r
Relation Encoder
Sentence Encoder
h_1^x h_2^x h_3^x h_4^x h_5^x h_6^x
x_1 x_2 x_3 x_4 x_5 x_6
Dual-Attention Decoder
s_t
Attention
Attention
c_t^r
c_t^x
c_t
softmax
y_t
GRU
c_{t-1} s_{t-1} y_{t-1}
Context Selection
MLP
c_t^x c_t^r

Figure 4: Dual attention Cao et al. (2017)

The entire architecture is build upon the Transformer Vaswani et al. (2017) which yields state-of-the-art accuracy using a stacked multi headed attention and fully connected layer architecture with an encoder and decoder. Additionally the input is sequentially encoded with a positional embedding removing the need for RNNs entirely.

One of the major battles with neural networks is how side and additional information is encoded. Narayan et al. (2017) have shown that using sentences $s_1..s_n$ along with context information $c_1..c_m$ directly fed into the attention mechanism during training, improves the presence of the contextual information. This is very similar to cue words, but within a neural architecture, making side information like real world knowledge, metadata or images available during training. It is not heavily studied yet, how to integrate contextual information in clever ways. Since the attention mechanism shifts the used words into the direction that are most likely to come next, it is possible to point to different words based on external knowledge. How external knowledge can be actually considered during attention is an open problem.

Documents are often structured into topical paragraphs. Common human based summarization actually describes summarization as chunking the text into global keywords, segments consisting of paragraphs or sentences, that are aligned with a topics. These topics form natural clusters and emphasize different arguments within the source document. Wang et al. (2018) introduce a topic aware summarization model, splitting the encoder into a word and topic encoder. The topic encoder learns topics pre initialized with *LDA* topic model distributions. Both encoders are then glued together via a biased probability generation function - a softmax with a linear layer. While not entirely new, they leverage reinforcement (teacher) learning used in Paulus et al. (2017) and pointer mechanisms with coverage like in See et al. (2017) yielding state-of-the art results.

2.2 Multi-Document Summarization

Many techniques applied to document summarization are also valid for multi-document summarization. A key difference is the preprocessing, since grouping documents into clusters is a task in itself. Cao et al. (2016) propose a model training a convolutional neural network (*CNN*) for text classification, using the resulting weights to map sentences to their categories. This in turn puts special emphasize on words belonging to the categories. Each document yields a single summarization, resulting in a cluster of single document summaries. They successfully leverage annotated labels that belong to broader categories.

Yasunaga et al. (2017) combine a sentence relation graph and sentence embeddings to initialize their model. The graph is an approximate discourse graph (*ADG*) for sentences that uses discourse markers, events, entities and noun references. Yasunaga et al. (2017) improved upon the *ADG*, creating a personalized discourse graph (*PDG*) using personalization features such as position in the document, no. of nouns, sentence length or co-reference verbs. As can be seen in figure 5 the relation graph and the learned sentence embeddings are chained with a *CNN*. The sentence embeddings are learned via *GRU* units.

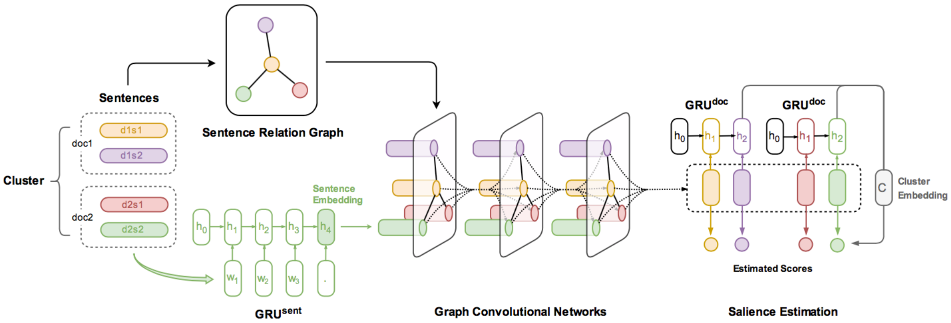


Figure 5: Graph-based neural multi-document summarization Yasunaga et al. (2017)

In the end, salience estimation outputs salience scores for each sentence. Instead of simply sorting by scores Yasunaga et al. (2017) fabricated a heuristic, by using cluster embeddings, trained with a *GRU*, over all documents to get a global view of all documents. Through the salient score the top sentences are selected in combination with an attentional softmax via the cluster embedding and *tf-idf* to validate if the current sentence is needed in the current summary. The lower the information density of a sentence the lower the chance that it is used in the target summary.

Since *Wikipedia* is one of the best known and free online encyclopedia, Liu et al. (2018) successfully build their summarization system on the massive amount of curated data. Treating *Wikipedia* as a multi-document summarization system they crawled all articles that conforms to the style guide and cited sources, as well as enhancing each article headline with the top 10 Google search results. They then summarized different topics such as “Machine Learning” with all the related documents. Processing those documents is a data heavy task and mostly suited to simpler models like the transformer by Vaswani et al. (2017) needing a fraction of the parameters and training costs than comparable baseline models. Liu et al. (2018) used the *Wikipedia* articles as summaries of broader concepts such as “Machine Learning”. They crawled all validated referencing articles that are categorized as “Machine Learning” concatenating a huge document collection. To process such large documents they used a memory compressed and local attention mechanism that increased the process-able sequence length by 3x.

2.3 Embeddings and Language Models

Document summarization is specialized task within the *NLP* community. Embeddings and language models organize text into probabilistic and contextual models. For years it was a common approach to initialize all text based models with unsupervised word embeddings, using shallow neural networks with one layer. They commonly optimize the following language model objective

$$P(w_t|w_{t-1},\ldots,w_{t-n+1}) = \prod_{t=1}^{n} P(w_t \mid w_1,\ldots w_{t-1}) \tag{1}$$

The most basic language model computes the conditional probabilities given a word w_t and the preceding words $w_1,\ldots w_{t-1}$ using the chain rule. To compute this function with respect to each word is equivalent to maximizing the cross-entropy loss of the softmax, a categorical non-linear function summing to one.

$$P(y = j \mid x) = \frac{exp(x^T \cdot w_j)}{\sum_{t=1}^{|V|} exp(x^T \cdot w_t)} \tag{2}$$

Up to 2017 the most used simplified objective function is skip-gram with a sliding window over words as context, replacing the simple softmax with.

$$\frac{1}{T}\sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log P(w_{t+j}|w_t) \tag{3}$$

Due to inefficient training of the softmax and descending objectives such as skip-gram, using hierarchical softmax or negative sampling with noise contrastive estimation is the only valid alternative. To name the most important models, which are often pre-trained on *Wikipedia* and available for download: *Word2Vec* by Mikolov et al. (2013), *GloVe* by Pennington et al. (2014), *Doc2Vec* by Le and Mikolov (2014), *Dep2Vec* by Levy and Goldberg (2014a), *Dict2Vec* by Tissier et al. (2017) and *FastText* by Bojanowski et al. (2016); Joulin et al. (2016).

Since language models are generically trained on *Wikipedia* it is possible to use other languages such as German. Theoretical investigations have lead to a few conclusions about word embeddings. Levy and Goldberg (2014b) have found that *Word2Vec* is essentially a positive point-wise mutual information (PPMI) matrix, where each word has assigned probabilities that are strongly associated with certain dimensions, that represent contexts. Which means that it is entirely possible to find an optimal setting that is as accurate as a word embedding with classical algorithms.

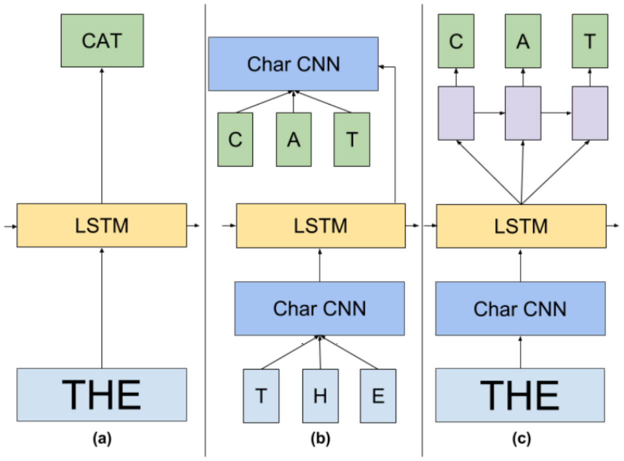


Figure 6: Character *CNN* and *LSTM* strategies Józefowicz et al. (2016)

Józefowicz et al. (2016) gave an overview of the different current language models building on character level embedding models, often pre-trained on word embeddings. Character based representations of words are itself d dimensional vectors for each character, belonging to the n-grams of a word. The average of all characters is the vector of a word. They explore different architectures for language modelling, comparing three different models with differing inputs. As can be seen in figure 6 the strategies are plain *LSTM* with word inputs and softmax as output, *CNN* consuming single character inputs and outputs and a character *CNN* with a softmax output. In theory all work reasonably well, but character *CNN* on both ends work best.

McCann et al. (2017) introduced one of the first language models with the idea of a working transfer learning procedure. They trained an attentional sequence-to-sequence model for neural machine translations *MT-LSTM*, learning sentence and word level abstractions. The model was a standard two-layer, bidirectional long short-term memory, on a source language (English) to a target language (German) achieving state-of-the-art performance. Since we are interested in the transferable aspects that generalize to different problems, the weights of the encoder are kept and later used during training for the specialized task. In order to update the gradients of the encoder h it needs to run the full model deriving the following formula

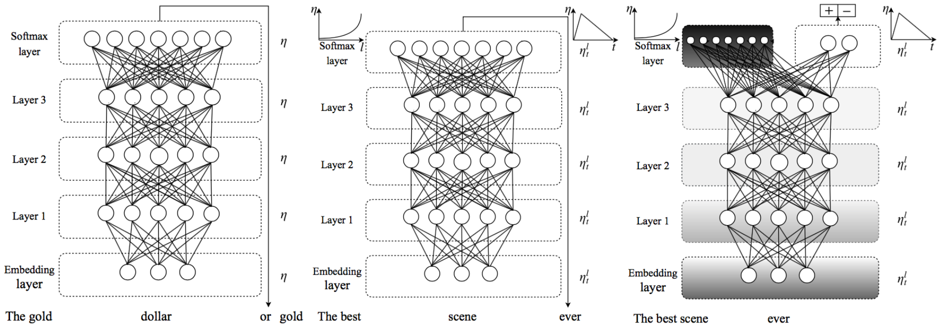
$$\begin{aligned} \textit{CoVe} &= \textit{MT-LSTM}(\textit{GloVe}(w)) \quad (4) \\ \widetilde{w} &= [\textit{GloVe}(w); \textit{CoVe}(w)] \quad (5) \end{aligned}$$

The idea behind this is, that higher level features can be transferred, learning generic features in sequence-to-sequence tasks to downstream tasks. By first using *GloVe* on the word-level and then the *MT-LSTM* we are creating layers of abstractions. While this is standard practice in computer vision tasks with pre-trained *CNN* it is a relatively new concept in *NLP*. Recent advances by Peters et al. (2018) have shown that it is possible to generalize the *CoVe* model, without training on a machine translation task, which require source and target language pairs. This is unnecessary and hard to organize, especially in an environment with multiple languages. The model is called *ELMo* and it optimizes the following language model.

$$\begin{aligned} &\sum_{k=1}^{N}(\log p(t_k \mid t_1, \ldots, t_{k-1}; \theta_x, \vec{\theta}_{\text{LSTM}}, \theta_s) \quad (6) \\ &+(\log p(t_k \mid t_{k+1}, \ldots, t_N; \theta_x, \overleftarrow{\theta}_{\text{LSTM}}, \theta_s)) \quad (7) \end{aligned}$$

Where $\theta_x, \vec{\theta}_{\text{LSTM}}, \theta_s$ are the parameters of the input embeddings x, weights of the language model $\vec{\theta}_{\text{LSTM}}$ in both directions and θ_s the output of the softmax of the language model, maximizing the log likelihood of tokens before and after the current token. The *ELMo* model is then used in combination with a task RNN, where the task specific input x is concatenated with the *ELMo* vector v. Both are jointly trained in combination with a final linear layer which is then fed into a specific downstream task. The problem though is that *ELMo* needs some adaptations given the architecture.

Shortly after Howard and Ruder (2018) have improved on *ELMo* with the *ULMFit* model. Universal language model fine tuning is the process to use inductive transfer learning to pre-train on large document databases and then refine the model with domain specific knowledge, shifting the parameters into the target domain. The model has three stages as can be seen in figure 7



(a) LM pre-training (b) LM fine-tuning (c) Classifier fine-tuning

Figure 7: ULMFit model Howard and Ruder (2018)

First, general pre-training of a language model using *Wikipedia*. Second using the pre-trained language model and fine tune on the target task. Different layers exhibit different features and need different fine tuning to adapt parameters. Therefore discriminative fine-tuning, e.g. different learning rates for each layer and slanted triangular learning rates, that linearly increases and then decays learning rates to converge fast and be adaptable later. Third classification (or any target task) fine tuning with gradual unfreezing, e.g. training only one layer at a time, adding more over time, to tackle catastrophic forgetting.

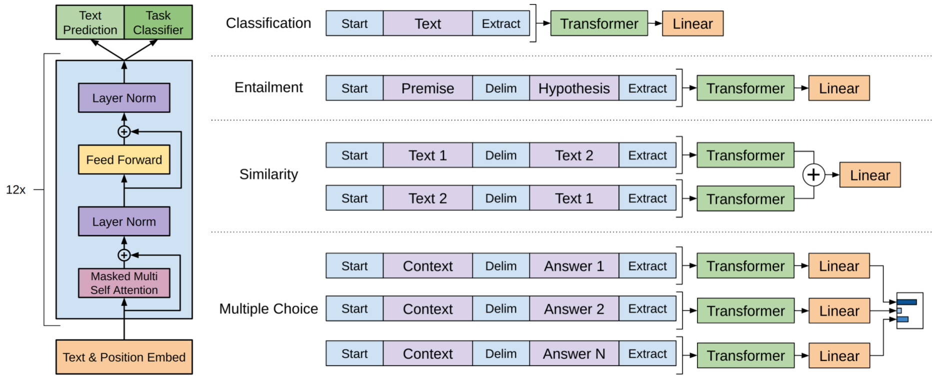


Figure 8: Transformer language model Radford (2018)

A third interesting way to train language models emerged shortly after. As depicted in figure 8, Radford (2018) propose a transformer language model.

The goal is to train a high-capacity language model leveraging the Vaswani et al. (2017) model. After training, there is a fine tuning stage for each input given the different tasks using structural encoding proposed by Rocktäschel et al. (2015). Each input x with a label y can be generically fed into the transformer language model chained with a linear layer and softmax. This generic representation makes all tasks with a supervised learning goal feasible.

3 Outlook

In conclusion this work presented the major aspects and theoretical foundations of document summarization systems and modern language modelling. Further it defined the problem space, problems to tackle and state-of-the-art work. While document summarization is a first step in creating such systems, the ultimate goal is to set the preliminaries for a multi-document summarization system. The rise of modern language modelling approaches gives new up speed into this particular field of research. However most modern day systems are not quite there yet. A lot has to be done and improved, especially if the summaries should be fluent, readable and concise. Future directions must find new ways to include real world knowledge, contextual information and structural/logical formulas. Something not discussed in this paper is the possibility to extend this work to a dossier generation system.

3.1 Dossier Generation

While this work is primarily focused on document summarization, a larger goal would be to create automated and semi-automated dossiers. A dossier is a collection of papers or other sources, containing detailed information about a particular subject. Essentially an information retrieval/recommender system based on keywords/user input in combination with multi-document summarization. Haelker (2015) identified the following aspects of a dossier:

1. Designed to fit a (topical) narrative / problem definition
2. Chronological / Historical / Hierarchical
3. Transparent and comprehensible
4. Presenting central (non-biased) arguments
5. Shallow at first glance, deep at second look

Since multi-document summarization is a hard problem, dossiers will be a suitable goal for further research. As much as summaries are a meta format for text and collections, dossiers are a meta format for clusters of collections and sources.

4 Acknowledgements

We thank Prof. Dr. Kai von Luck for the guidance throughout the seminars and resources provided by the CSTI Lab.

References

Aggarwal, C. C. and Reddy, C. K., editors (2013). *Data Clustering: Algorithms and Applications (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series)*. Chapman and Hall/CRC, 0 edition.

Allahyari, M., Pouriyeh, S. A., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., and Kochut, K. (2017). Text summarization techniques: A brief survey. *CoRR*, abs/1707.02268.

Blei, D. M. (2014). Build, compute, critique, repeat: Data analysis with latent variable models. *Annual Review of Statistics and Its Application*, 1(1):203–232.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3(4-5):993–1022.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *CoRR*, abs/1607.04606.

Cao, Z., Li, W., Li, S., and Wei, F. (2016). Improving multi-document summarization via text classification. *CoRR*, abs/1611.09238.

Cao, Z., Wei, F., Li, W., and Li, S. (2017). Faithful to the original: Fact aware neural abstractive summarization. *CoRR*, abs/1711.04434.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. In *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, volume 41, pages 391–407.

Gong, Y. and Liu, X. (2001). Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 19–25, New York, NY, USA. ACM.

Haelker, N. (2015). Teilautomatisierte erstellung von dossiers auf der basis von textmining-verfahren.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *ACL*. Association for Computational Linguistics.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759.

Józefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., and Wu, Y. (2016). Exploring the limits of language modeling. *CoRR*, abs/1602.02410.

Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. *CoRR*, abs/1405.4053.

Levy, O. and Goldberg, Y. (2014a). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 302–308.

Levy, O. and Goldberg, Y. (2014b). Neural word embedding as implicit matrix factorization. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc.

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. (2018). Generating wikipedia by summarizing long sequences. *CoRR*, abs/1801.10198.

McCann, B., Bradbury, J., Xiong, C., and Socher, R. (2017). Learned in translation: Contextualized word vectors. *CoRR*, abs/1708.00107.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

Nallapati, R., Xiang, B., and Zhou, B. (2016). Sequence-to-sequence rnns for text summarization. *CoRR*, abs/1602.06023.

Narayan, S., Papasarantopoulos, N., Lapata, M., and Cohen, S. B. (2017). Neural extractive summarization with side information. *CoRR*, abs/1704.04530.

Paulus, R., Xiong, C., and Socher, R. (2017). A deep reinforced model for abstractive summarization. *CoRR*, abs/1705.04304.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proc. of NAACL*.

Radford, A. (2018). Improving language understanding by generative pre-training.

Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kociský, T., and Blunsom, P. (2015). Reasoning about entailment with neural attention. *CoRR*, abs/1509.06664.

Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. *CoRR*, abs/1509.00685.

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

Tissier, J., Gravier, C., and Habrard, A. (2017). Dict2vec : Learning word embeddings using lexical dictionaries. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 254–263.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

Wang, L., Yao, J., Tao, Y., Zhong, L., Liu, W., and Du, Q. (2018). A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. *CoRR*, abs/1805.03616.

Yasunaga, M., Zhang, R., Meelu, K., Pareek, A., Srinivasan, K., and Radev, D. R. (2017). Graph-based neural multi-document summarization. *CoRR*, abs/1706.06681.