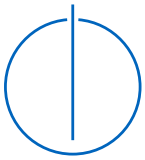
DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Multi-Task Deep Learning in the Legal Domain

Christoph Gebendorfer



DEPARTMENT OF INFORMATICS

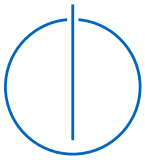
TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Multi-Task Deep Learning in the Legal Domain

Multi-Task Deep Learning im Rechtsbereich

Author:	Christoph Gebendorfer
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Ahmed Elnaggar, M. Sc.
Submission Date:	12.07.2018



I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 12.07.2018 Christoph Gebendorfer

Acknowledgments

In front, I want to express my deepest gratitude to my advisor Ahmed Elnaggar who supported me throughout this work and feel very fortunate that i could expand my horizon with his help in this area of research.

I want to thank Prof. Matthes for the opportunity to write this thesis at his chair Software Engineering for Business Information Systems (sebis) in cooperation with msg systems AG.

A special thanks goes to Rainer Singvogel and Nicole Ondrusch from msg systems AG who gave me the opportunity, support and trust to work in this special area.

In addition, I want to thank all the colleagues from the chair and msg systems AG who accompanied me on my way for creating such a positive environment while continuously backing me up.

Finally, I want to thank my family and friends for always having faith in me.

Abstract

The revival of deep learning yielded astonishing results in many tasks from computer vision, machine translation to speech recognition in the last years. This advancement is favored by the increasing availability of datasets and computational resources. On the other side, the legal domain with its serious demand for natural language processing applications cannot benefit in equal measure from it, since appropriate preprocessed legal datasets are highly limited or barely exist at all. In contrast to using datasets from other domains, we propose the usage of multi-task deep learning in order to exploit task-independent commonalities and overcome the dataset shortage in the legal domain.

As part of this work, we have created six different legal corpora for translation, text summarization and document classification. Five out of the six corpora descend from the DCEP [1], Europarl [2] and JRC-Acquis [3] corpus provided by the European Union which we processed for the immediate use with neural network based models. The last corpus is a collection of 42k documents containing court decisions of the seven federal courts of Germany scraped from their official website.

Based on these newly created corpora, various multi-task combinations within a task family (e.g. only translation tasks) and across task families (e.g. translation, summarization & classification) were trained on the state-of-the-art multi-task deep learning model, the MultiModel [4]. In addition, we compared the single & multi-task performance of the MultiModel on two different sets of hyperparameters to the state-of-the-art translation model, the Transformer [5]. The MultiModel trained on joint tasks is on an equal footing with the Transformer. We show that multi-task deep learning is advisable in situations where training data is sparse through experiments in which a jointly trained MultiModel is able to outperform a single-task trained MultiModel and the Transformer. Surprisingly, a combination across task families surpasses several combinations within task families. Finally, we trained a combination which beats the JRC EuroVoc Indexer JEX [6] in the German multi-label classification task by nearly 14 points on the F1 metric.

Contents

1. Motivation

1.1. Motivation

Machine Learning moved into the focus of researchers and practitioners in recent years. Seemingly easy tasks for humans, which could not be solved effectively by computers are now tackled and solved with machine learning tools in tow. A subfield distinguished itself as very promising, yielding astonishing results for challenges across different areas, including computer vision, natural language processing, robotics, medical applications and data mining. This is *deep learning*, a machine learning discipline, which draws its capabilities from artificial neural networks. These neural networks learn from experience and represent their learned knowledge in the interaction of interconnected components. By stacking these components, a deep architecture is created posing the eponym for this exceptional innovative field. Apparently, artificial neural networks did not fell out of the blue. In fact, the first artificial neural network was already developed in 1962 [7]. The training of the same was effectively solved with the *backpropagation algorithm* introduced in 1986 [8]. More complex network structures including convolutional neural networks (CNN) [9] which perform intuitive human tasks such as object recognition were developed afterwards. However big breakthroughs would still take some time in coming. Computing power to speed up the training process accordingly, rose over the last decades, reaching a certain threshold lately, finally enabling the full potential of deep learning. Especially developments of faster and more dense *graphical processor units* (GPU) contributed. Manufacturers even specialized in producing hardware with particular interfaces [10] for deep learning requirements. Rightly, deep learning ushers a new era for solving ostensibly impossible problems.

However, whereas deep learning models beat record after record and win numerous contests in pattern recognition and machine learning, the propagation of these models across industries is far from pervading all levels. It is the current challenge to apply these models, induce new processes and support humans in their work. Hence, domains move into focus which are truly based on tasks in which deep learning shines and can largely benefit from its potential. This certainly includes law and the *legal domain*. A large proportion of legal professionals are being confronted with tasks of natural language processing every day. Ever since laws have been designed and politics been operated, all associated acts had to be documented precisely. This leads to an exceptionally large text base, which is growing steadily. Instead of manually handling the paperwork, deep learning can assist or completely carry out processing it. Work in

this direction has already been conducted, e.g. translating legal documents [11] or even classifying verdicts of the French Supreme Court [12]. Nonetheless, a large amount of possibilities are not exploited yet. Our motivation lies in exploring these and advancing the application of deep learning in the legal domain.

1.2. Problem Statement

Two factors play an important role when applying deep learning in the legal domain.

Computational Power and Tools The *parameter* count for sophisticated neural networks is in the hundreds of millions. Depending on the used model, it can even be higher. The training process involves repeatedly updating these parameters through the backpropagation algorithm. In order to efficiently execute the training, potent hardware is needed. Also, interfaces and tools to rapidly implement models may not be missing.

Large Datasets The provision of a large dataset is key to good performance of a deep learning model. A rule of thumb is that a *supervised deep learning* algorithm should have at least 10 million examples to match or exceed human performance. Each sample needs to be labeled appropriately to the task at hand. Therefore, large annotated datasets are indispensable.

The introduction of parallel training methods and the usage of GPU hardware [13] accelerated the training speed in the last years. Adequate computational power is no longer a problem to machine learning. The gradual decline of Moore's Law did not inhibit growth. Parallel computing won significance, which appositely aligns with concurrent training. Concluding, computational power is not a problem to deep learning in the legal domain. It is independent from the legal domain.

The real difficulty lies in acquiring annotated datasets. A labeled dataset is essential to training a model on a specific task. The creation of annotated datasets is thriving to support improvements in general tasks, such as object recognition [14], machine translation [15] or speech recognition [16]. However, these datasets include samples across domains and do often not suffice for acceptable accuracy in special domains such as the legal domain. Against the huge amounts of text, which are available in the legal domain, only a small proportion is publicly available and labeled appropriately. This leads to the following problem:

Annotated legal datasets are highly limited or barely exist at all.

Many legal tasks, including named entity recognition, named entity disambiguation, question answering, text summarization, document classification, part-of-speech tagging, semantic analysis and taxonomy generation are in desperate need of preprocessed

datasets. The only exception to this dataset shortage is the legal translation task, which is supplied with multiple huge datasets 4.1.

We try to counteract the data shortage with the application of multi-task deep learning in the legal domain. Multi-task deep learning describes the training of multiple tasks on one model in order to mitigate data scarcity and establish *transfer learning*. Hence, the overall goals of this work are:

- Exploit commonalities and overcome task-specific dataset shortage in the legal domain
- Establish transfer learning for better results in legal text tasks
- Support generic and task-independent deep learning architectures

1.3. Research Questions

We are going to answer the following research questions in regard to the application of multi-task deep learning in the legal domain.

1. Can multi-task deep learning be beneficial for tasks in the legal domain?
2. How does training simultaneously on multiple tasks of the legal domain compare to training on each task separately?
3. How far is multi-task deep learning from *state-of-the-art* solutions in the legal domain?
4. What needs to be considered for choosing suitable *hyperparameters* for multi-task deep learning in the legal domain?

1.4. Thesis Contribution

In the course of this work, two major contributions are made.

Legal Corpora for Translation, Summarization and Classification We provide preprocessed datasets for the legal tasks. Each *corpus* contains a corresponding test set, which allows the evaluation and comparison of different models. We created three ready-to-use corpora for the legal translation of 21 language pairs. In addition, we compiled a corpus for legal summarization and multi-labeling, which encompasses the inference of a document title and content labels from the document paragraphs. Finally, we scraped german court decisions from the public website and assembled a corpus for classifying the originating court and verdict outcomes. All corpora are made publicly available (see 4.15).

Integration with State-of-the-art Model We contribute to the integration of legal tasks with Tensor2Tensor [17]. For each conducted task, we implemented respective data generators for Tensor2Tensor. This allows researchers to easily switch models on these tasks and build upon our efforts. The data generators are also made publicly available.

1.5. Research Milestones

We followed a stepwise approach to answering the research questions and contributing to a solution for the data scarcity in the legal domain. The first milestone included the analysis of the MultiModel and a stocktaking of existing legal datasets with their viability for this work by investigating related work in multi-task deep learning and natural language processing in the legal domain. Subsequently, all resources needed to be assessed, leading to the compilation of corpora for the usage in translation, summarization and classification with the state-of-the-art multi-task model. The integration into Tensor2Tensor constitutes the third important milestone to our work. On top of this basis, we conducted numerous experiments. The training of the MultiModel with the assembled corpora across the three legal tasks serves to generate information according to our hypotheses. Finally, we conclude from the evaluation of the training results and answer the research questions.

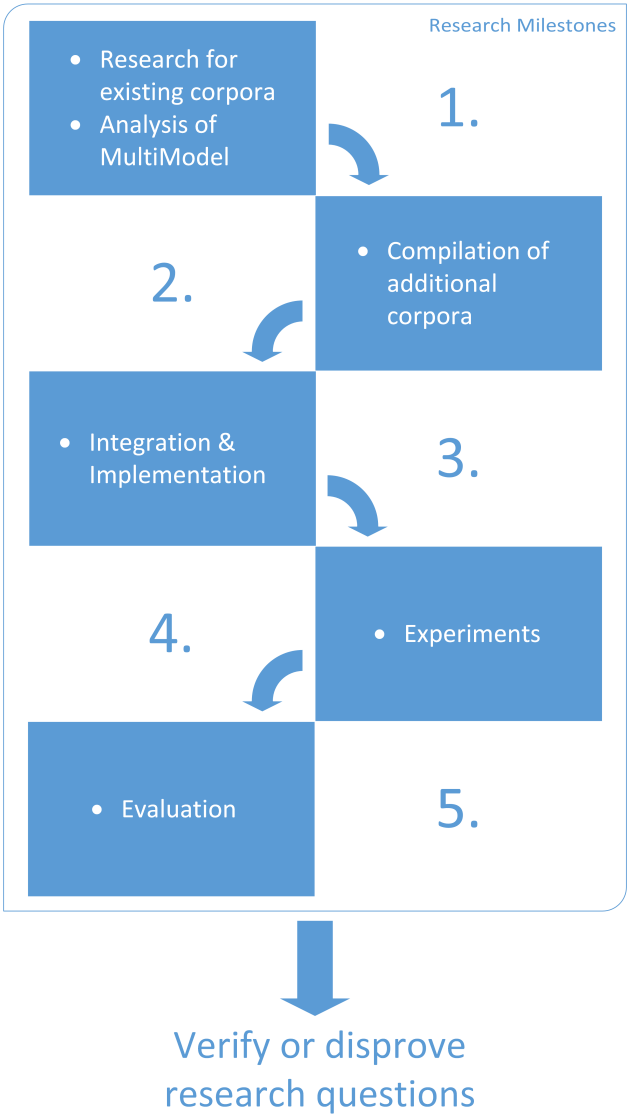


Figure 1.1.: Research Milestones

1.6. Outline

The thesis is divided into three major parts. The first part starts off with foundations in natural language processing, deep learning and its multi-task subfield, followed by the reference and discussion of related work. Subsequently, existing legal corpora and the newly compiled datasets are presented in addition to the implementation work regarding the integration of the legal tasks into Tensor2Tensor. After the middle part, numerous experiments and their evaluation results are discussed. Finally, we give conclusions to and an outlook from our work.

Figure 1.2.: Thesis Outline

2. Foundations in Natural Language Processing and Deep Learning

Before diving into the core of this work, foundations in natural language processing and deep learning are given. These two fields gained massive traction recently and keep growing in light of groundbreaking achievements. Opportunities of natural language processing in the legal domain and multi-task deep learning are part of the ensuing overview.

2.1. Natural Language Processing

The field of *natural language processing* (NLP) encompasses in a broader sense every transformation or interpretation of human languages through computers. Technologies using NLP are present everywhere, for example in the input correction during typing on the phone, search engines trying to analyze text on websites in order to provide better results or speech recognition software transforming spoken word to its textual representation. All these tasks are challenging, since natural languages do not follow strict rules like programming languages while also steadily evolving in expressiveness. In contrast to artificial languages, utterances in human languages can express the same meaning while having a different syntax as well as featuring deviant semantics with the usage of the same word. Yet, human languages posses underlying structures, syntactic rules - grammars to form respective sentences. Capturing these patterns, inferring their semantic, interpreting them correctly and transforming them accordingly lies at the heart of natural language processing. Within this work, various tasks were selected for the application to legal text corpora. This includes translation, summarization and classification. In the following, each task is presented.

2.1.1. Translation

As part of this work, the translation task describes the process of correctly transferring the semantic of a *medium* from one language to another. In NLP, translation is generally carried out on a sentence-to-sentence text basis. This allows a more accurate translation of larger texts and whole documents compared to a sole word-by-word dictionary approach. However, a sentence-to-sentence translation approach is also more difficult, because the semantic of a sentence is highly dependent on the composition of its words. Hereby, difficulties arise in the realization of this varying semantic across languages.

In German, it is often the case to connect two words to form a compound word with related meaning. One may use another word or two words in English to express the same meaning. For example, the German word 'Tagebuch', which is a compound noun of 'Tage' (days) and 'Buch' (book), may be translated to 'diary'. A translation model needs to be aware of these patterns. In a different context 'Tagebuch' may be translated to 'journal'. It could even be translated to 'blog' in the Internet context. As follows, it is likely to encounter correct translations of one sentence in various forms. On the other hand, a dictionary approach would constantly yield the same word or propose multiple words out of non-decisiveness due to missing context.

As mentioned, the key to flawless translation is context and the true understanding of a sentence as a whole. While machines are still not capable of generally understanding the deeper meaning of sentences, they are pretty good at recognizing and memorizing patterns. Therefore, corpus-based work is common in translation. That means a large dataset is used to train or adjust translation systems. Like humans learn additional languages by practicing with examples, computers do learn translating from one language to another by seeing a large numbers of aligned samples. This way, the machine can become context-aware to a higher degree and learn correct translations besides the difficulties as stated previously. According to Stephen R. Anderson, there exist over 6800 languages in the world [18]. Each language has its own ways of expressing our surroundings and can highly vary in written form. e.g. using different symbols. Without a doubt, this makes translation one of the hardest tasks in natural language processing and keeps researchers eager to develop new translation models and approaches to yield improvements.

Multilingual resources gained significance in the face of international collaboration. Growing political and economic relationships bring attention to legal translation. These circumstances lead to a rising demand for automatic translation in the legal domain. Especially, international unions have common resources, which need to be available in the languages of their partners. The European Union, North Atlantic Treaty Organization or United Nations are good examples for such alliances. Naturally, the amount of text one organization has to administer increases with the language diversity across the member states. At present, professional translators provide these translations. That also has a good reason. Deviations with a change in the information in these translations could be fatal. Since the translations are not only bound to a general transmission of the semantic, it is inevitable to communicate the exact same facts with absolute fidelity [19]. Legal translation comes with its own characteristics and has to be treated carefully, as Garzone [20] described:

> Legal translation is certainly among the varieties of translations where the translator is subject to the heaviest semiotic constraints at all levels: the language of the law is typically formulaic, obscure, archaic; legal discourse is culturally mediated; legal texts have a special pragmatic status.

Obviously, translation in the legal domain creates new challenges. This gives us the incentive to select it as a major task that we want to investigate as part of this work.

2.1.2. Summarization

Summarization is commonly executed on text, but may be applied elsewhere, e.g. on speech recordings with the same techniques [21]. We define summarization as the reduction of text to a smaller form which conveys the same message. Text summarization is technically a text-to-text conversion similar to translation. In contrast, the input and output is in the same language while their length differs. Hereby, the objective is filtering out the essential meaning of a text and reflecting it in a short version. There are two forms of summarization.

First, the extractive summarization task concentrates on finding the most important segments of a text and depicts them with the same words as they appeared in the original text. This kind of summarization mostly preserves whole sentences and thus produces syntactically correct summaries. Nonetheless, they often appear incoherent due to leaving out connective sentences or structures. Alternatively, the abstractive summarization is a strategy to paraphrase the original text into a new concise version. In this variant, used words are chosen freely and do not have to exist in the source at all. While abstractive summaries come in a more organic nature, the challenge is not to falsify original statements. By leaving out negations or important entities, the meaning of a summary can highly deviate.

Summarization is likely a corpus-based task. E.g. transcribing a news article body to its lead statement or extracting a product title from its description. The use cases for this NLP task are manifold. A lot of reading work and time by humans can be saved through effective summarization. This incentives linguists to engineer new approaches to text generation and information retrieval.

In particular, the legal domain can make great use of automated summarizers for documents. The amount of text in legal environments tends to be excessive. Law firms have to grapple with their internal document accumulation by reading through vast collections of case files to grasp essential information. Likewise, court processes are recorded in detail, whereas a short summary often satisfies to answer questions to outstanding persons. Reducing this effort by summarizing legal texts is desirable. However, the same conditions of legal translation apply to summarization. The nature of legal text complicates its processing. Summarization needs to be done carefully to not lose valuable information while reducing the text body. This makes the legal summarization task highly dependent on the corpus. Extractive summarization is the preferred way in the legal domain to preserve the syntax and information as best as possible. Our goal is to explore novelties by incorporating summarization into our multi-task deep learning undertakings.

2.1.3. Segmentation and Labeling

Text segmentation and labeling tasks are based on grouping text together or dividing it into classes. This includes *tokenization*, sentence splitting, named-entity extraction and chunking. On closer inspection, a distinction is made between how many classes can be predicted and how many classes are assigned to one example. In the following, a short overview of classification methods is given.

Unary Classification The objective of unary classification is to find examples of one class amongst the input. No statement can be made for examples which are not classified. Therefore, unary classification is also called one-class classification. Unary classifiers can be used for detecting natural language on images [22].

Binary Classification The binary classification is a common form of classification. Within this task family, a classifier divides the input into two classes. As follows, each example must be part of one class. For this reason, the classes are often labeled positive or negative for problems predicting the existence or absence of properties for one objective. Examples for binary classification tasks are the sentiment analysis of twitter posts [23] or the identification of malicious URLs [24].

Multi-class Classification The multi-class classification involves the classification over three or more classes. This way it is possible to divide the input into an arbitrary number of groups and classify examples much more precisely. For example, multi-class classification is applied in named entity extraction in order to find and assign a word to a specific entity class [25], but also in translation models for predicting the next word over a fixed predefined vocabulary [26, 27].

Multi-label Classification In multi-label classification or often called multi-labeling, each class is considered as a binary classification problem. Therefore, each input example can get assigned to multiple classes. This is very useful in situations, in which correlations between classes exist. Modeling such a problem with a multi-label classification task allows the exploitation of these relations. Multi-labeling examples are the labeling of emails [28] or the classical document labeling [29].

Legal professionals can be greatly supported by automated classifiers in text segmentation and document labeling. Even better, users with no legal education can greatly benefit from systems which take over legal text segmentation and labeling tasks such as estimating the outcome of a verdict or giving the likelihood of an accepted petition for naturalization. It is evident that there exists a lot of text in legal environments. Getting control over it needs effort. It is our objective to bring legal classification forward by enclosing it in out multi-task deep learning experiments.

2.2. Deep Learning

Deep learning is a machine learning discipline that allows computers to learn from experience and grasp their environment through a hierarchy of concepts. This way, a bundled specification of the knowledge base is omitted. Higher concepts are represented by its relations to simpler concepts. These are technically realized through layers in an interconnected graph known as artificial neural network. Thereby, the stacking of numerous layers leads to a deep architecture which is the eponym for this exceptional field. The goal of an artificial neural network is to approximate some function f^* usually in regard to a classification or regression task. In the following, an overview of different types of neural networks is given. The depicted networks are not exclusive and parts often function as *computational blocks* for more sophisticated combined models.

2.2.1. Artificial Neural Networks

Deep Feedforward Network

The most used deep learning architecture is the deep feedforward network (FNN), also called multilayer perceptron or feedforward neural network. This type of network approximates a function f, which maps an input x to an output y in face of learned parameters θ.

$$y = f(x, \theta) \tag{2.1}$$

This architecture is called feedforward, since the information is propagated only in one direction. Starting with the input x, the information flows through the computations which constitute the function f to the output y. The function f is usually a composition of intermediate functions $f^{(i)}$ where each function is applied subsequently. This results in a chain structure of functions $f^{\circ i}(x)$. Each function $f^{(i)}$ corresponds to a layer of the network. See figure 2.1 for an example with three hidden layers.

Convolutional Neural Network

Convolutional neural networks (CNNs) [30] are special neural networks which are best-suited for processing data of grid-like structures. This includes time-series data, where the sequence of data points can be seen as 1D array, 2D data such as pixels of image or spatial simulation data in 3D. A mathematical operation called convolution is responsible for the name of these networks. Convolutional neural networks use the convolution operation in at least one of their hidden layers. The convolution operation is used to smooth input data by using sparse interactions and shared parameters. The receptive field of a convolutional network is usually smaller than the input which reduces the connections between input and output units. In addition, the convolutional

operation uses a parameter set at every position of the input. Sharing reduces the overall amount of parameters and contributes towards lower memory requirements and increased efficiency. Besides the convolution operation, a common layer composition in a convolutional neural network includes a non-linear and pooling function. These two functions are used to crop values and make the learning of representations invariant to small variations in the input, e.g. a slight rotation of an object on an image. It is common practice to stack multiple convolutional layers to subsequently redraft the feature map of the layers in order to learn a hierarchy of spatial concepts in multidimensional data.

Recurrent Neural Network

Much like convolutional neural networks are generally used for processing data arranged in a grid of values, recurrent neural networks (RNNs) [8] are applied to processing sequential data. While other network types can also handle sequential data, recurrent neural network are specialized in processing long sequences, such as time-series data or large text bodies. Furthermore, the input size can be variable instead of fixed length. Similar to convolutional networks, the core concept enabling recurrent neural networks is sharing parameters between different parts of a model. A generalization across multiple time steps would not be possible with separate parameters for each time step. Recurrent neural networks can be used in different forms, e.g. for producing an output at each time step or producing a single output after reading a whole sequence of input data. Moreover, RNNs are generally found in sequence-to-sequence models.

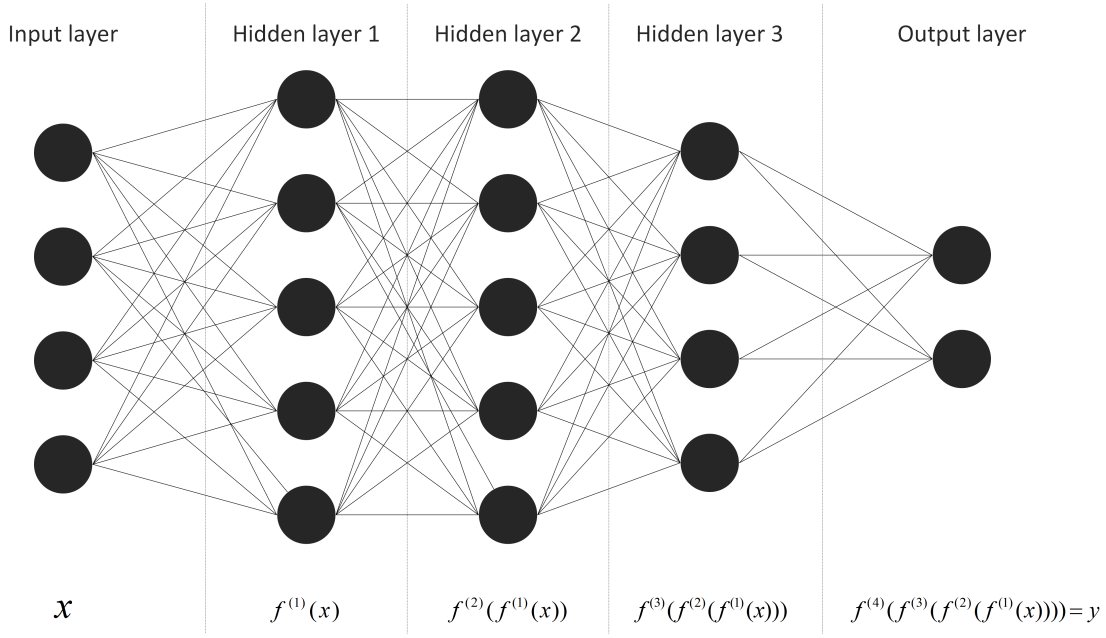


Figure 2.1.: Visualization of a deep feedforward network with three hidden layers

These models are also called encoder-decoder models in which the encoder takes the input sequence and produces a so-called fixed-length context. This representation is used as input to the decoder which produces the output sequence. The novelty of this approach is that input sequence and output sequence do not have to be of equal length. Likewise, it is possible to stack RNN layers [31]. Introducing depth in recurrent neural networks has been shown to improve the performance [32] by representing knowledge as a hierarchy of temporal dependent concepts.

Supervised Learning

Improving the approximated function f^* equates to the network learning the required knowledge to solve the underlying task. Teaching knowledge is done through training the network. During training, examples are fed into the model. Each example states a tuple (x, y) of an input x and expected output y. In doing so, the input x of the tuple is laid out on the input layer of the network which produces the output y^* on the output layer. Apparently, the output y^* should be next to the expected output y. The difference between y^* and y states the error between the actual and expected result. Subsequently, this error is propagated back through the hidden layers in order to adjust the parameters θ with the help of an optimizer. This procedure is called the backpropagation algorithm. As follows, the training data does not determine the output of each hidden layer. Activation functions are used to compute the hidden layer values. It is the learning algorithm's duty to determine how to use these layers to approximate f^* as best as possible.

2.2.2. Single-Task Learning

It is common that neural network models are only used to perform one task by training them in isolation. A model trained on a single task is specialized in exclusively solving this task, since all learned parameters are adjusted towards the same. The network approximates a function from the inputs to one output. This training technique is called *single-task learning* (STL) and is most commonly used in practice. Figure 2.2 shows an example of two neural networks trained on one task each. Single-task learning is often deployed because of one objective - solving one task and improving the results of it based on one dataset. This approach works great for many different areas, however it does not lead to generalization across tasks. In case, one wants to perform related tasks, it is necessary to train a new model with new parameters which manifest the according knowledge. For example, a common model trained on an English-to-German translation task cannot be used to translate from German to English.

As a consequence, it would require to train two different models in order to perform both tasks. This is clearly against our intuition. A human which can translate from English to German can likewise translate from German to English. Evidently, the knowledge required for these translation tasks is highly related. E.g. the knowledge of

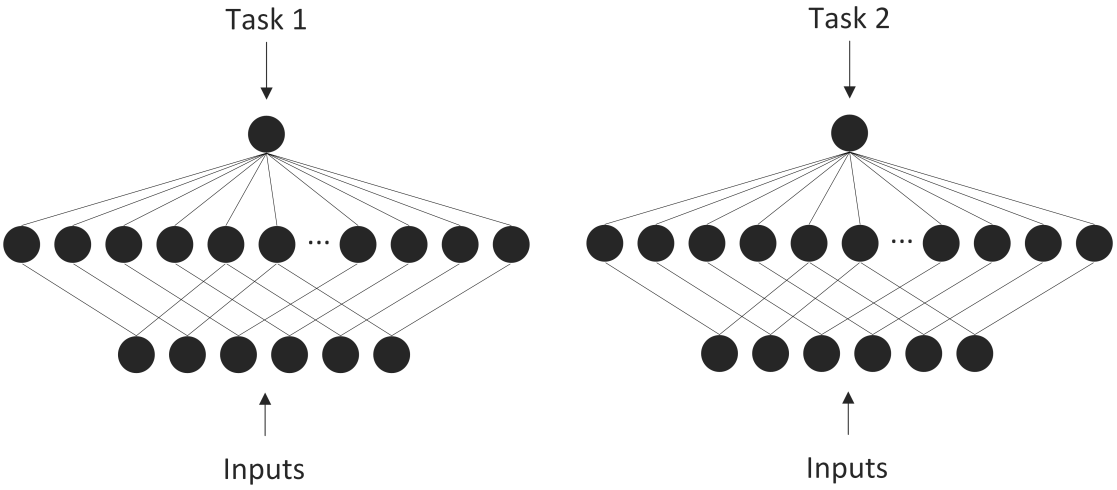


Figure 2.2.: Visualization of single-task learning with two artificial networks on two tasks

English grammar is adjuvant to understand an English sentence (English-to-German) as well as to compose an English sentence (German-to-English). Mastering translation includes the interpretation of sentences in both directions by transfer learning and creating interleaved knowledge bases which cannot be implemented through single-task learning. Hence, generalization suffers from training in isolation.

2.2.3. Multi-Task Learning

Promoting generalization across tasks is the goal of a technique called *multi-task learning* (MTL). With multi-task learning, it is possible for tasks to access internal representations established through other tasks by using common hidden layers (see figure 2.3). Multi-task approaches can be categorized into three different types.

One dataset, one input & multiple outputs There exists one dataset which contains multiple targets for each sample. The model has one input with multiple outputs.

Multiple datasets, one input & multiple outputs Multiple tasks are divided into multiple annotated datasets. The model contains one input which is used by training all tasks sequentially. The model provides respective outputs for the tasks.

Multiple datasets, multiple inputs & multiple outputs Multiple datasets are fed via multiple inputs into the model. Samples from all tasks are jointly learned through concurrent training leading to multiple outputs.

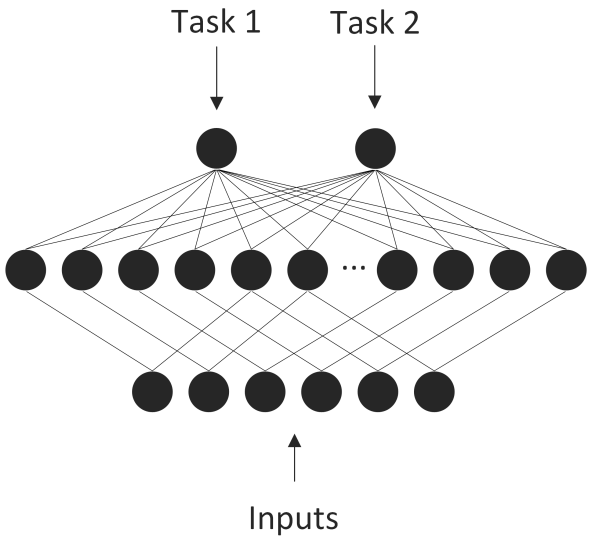


Figure 2.3.: Visualization of multi-task learning with one artificial network on two tasks

Multi-Task Mechanisms

Sharing learned parameters across tasks is the main idea of multi-task learning. The following mechanisms play a key role in multi-task learning and even allow performance gains for each single task when trained in a multi-task fashion.

Statistical Data Amplification & Attribute Selection Assuming that, a neural network is trained on two tasks T_1, T_2 with two datasets D_1, D_2 containing an independent amount of noise, then both tasks T_1, T_2 benefit from computing a shared representation layer L. As long as the neural network recognizes that the two tasks T_1, T_2 share the layer L, it can effectively learn this layer better by averaging the parameters over the varying noise in D_1, D_2. This also applies to attribute selection. The training of the single task T_1 with limited training data is somewhat difficult. A neural network will have problems attaching to relevant features and discarding irrelevant ones for a possible shared representation layer L. Training with more training data from multiple tasks allows the network to select attributes more accurately for the computation of the shared layer L [33].

Eavesdropping Suppose a hidden layer feature F that is beneficial for multiple tasks T_1, T_2 and easy to learn via training T_1. However during training T_2, it is hard to learn this feature, likely resulting in its neglection. Through learning both tasks T_1, T_2, task T_2 can eavesdrop on the hidden layer also trained by T_1. This kind of listening lessens the fade out of F by T_2 [34].

Representation Bias Neural networks are initialized with random parameters which are adjusted during training over time. Multiple training runs do not lead to the same learned parameters. Assuming that a task T_1 has two local minima a, b while task T_2 has the local minima a, c, they both share the minimum a, but do not share b and c. Single-task training of T_1 would equally lead to a parameter set ending in a or b. Instead training on both tasks T_1 and T_2, the learned parameters will generally end up in a. As follows, MTL favors representations which all tasks like and avoids representations that the majority of the tasks dislike [35].

Despite the advantages, multi-task learning is not universal. Multi-task learning has to be tested on problems to make sure that transfer learning actually happens in a positive way.

> MTL is a source of inductive bias. Some inductive biases help. Some inductive biases hurt. It depends on the problem. (Rich Caruana, 1997 [36])

The interesting field of multi-task learning facilitates transfer learning and poses a possible solution to dataset shortage.

3. Related Work

3.1. Legal Corpora

Our work connects with several points of research in the area of multi-task deep learning, natural language processing with neural networks and the legal domain. Driven by the need for large amounts of textual data, researchers propose new corpora which can be used for developing deep learning architectures and making fair comparisons between different models. Though, translation is the only task provided with numerous preprocessed corpora [37] predominantly originating from the legal domain [1, 2, 3, 38]. The WMT[1] and the OPUS project [39] are mainly responsible for making these corpora accessible. Other tasks are not supported in equal measure, especially in the legal domain. Rare work in this direction include Grover et al. [40], who assembled the HOLJ corpus for extractive summarization of British judgements. Though not from the legal domain, the CNN/Daily Mail corpus [41, 42] and the Gigaword dataset [43] emerge as prominent summarization corpora, while the smaller DUC task sets [44] fall into oblivion for the application with neural network based models. Test sets often originate from the same corpus, however different portions are drawn [45, 42, 46] for the evaluation with the ROUGE [47] metric. Alongside, Huber [48] compiled the Old Bailey Corpus which contains proceedings of the Old Bailey suitable for classification, language modeling and summarization. Qualitative corpora used in training models are often not made public or only accessible by paying a fee [43]. In addition, research in specific domains often excludes the usage of corpora from other domains. To our findings, translation experiments in other domains occasionally include training sets from the legal domain, usually the Europarl corpus [2], while test sets originate from either the IT domain[2] or news articles[3].

3.2. Natural Language Processing & the Legal Domain

Auto-regressive sequence models based on neural networks drive common text-to-text tasks and are omnipresent in natural language processing. Vaswani et al. [5] proposed the Transformer which represents the current state of the art in translation. With a BLEU [49] score of 41.8, the Transformer placed a new mark on the WMT14 English-to-French

[1] http://www.statmt.org

[2] http://www.statmt.org/wmt16/translation-task.html

[3] http://www.statmt.org/wmt14/

test set. Improvements and modifications are exhibited [50, 51] to further increase its translation performance. Furthermore, legal translation with neural network based models has not been specifically investigated, besides the realization with statistical machine translation systems [11]. The assessment of deep neural networks within other text-to-text problems than translation is not equally established due to the sparse availability of common task-specific datasets. Therefore, research proceeds on a wide range of problems often involving custom processed datasets. Against that, Sajjad et al. [52] show how corpora from different domains can be combined to increase translation quality by pretraining a neural net with out-of-domain data and fine tuning it with in-domain data. Rush et al. [45] initiated work on abstractive summarization with neural networks and induced researchers to continue with sequence-to-sequence models [42, 53, 46, 54]. Additional variants are proposed, including Liu et al. [55] with a GAN [56] approach, Liu et al. [57] with a hybrid approach of extractive and abstractive summarization as well as Hasselqvist et al. [58] with a query based variant. Again, legal summarization has not been carried out with neural network based models due to data scarcity. Earlier research in the legal domain concentrates on extractive summarization of court judgements [40, 59]. Opposing to translation and summarization, multiple work precedes in legal classification, especially multi-label classification [60, 61, 62, 63, 6] involving the *EuroVoc thesaurus*[4]. The EuroVoc thesaurus is a collection of over 6700 hierarchically organized domains and subdomains. Tasks include the assignment of multiple EuroVoc domains to legal documents of the JRC-Acquis [3] and partially overlapping Eur-Lex Database[5]. Steinberger et al. [6] achieved a respectable accuracy of 47.3% on German and 48% on English documents of the JRC-Acquis. Additional applications of deep neural networks in the legal domain are seldom. Alschner & Skougarevskiy [64] are using character-based recurrent neural networks to produce legal texts of specific classes. Wyner & Casini [65] show performance gains of deep neural networks in contract element extraction in comparison to linear classifiers. Further, Chen & Eagel [66] use a random forest approach to predict asylum seeking adjudications while Nejadgholi et al. [67] apply a combination of word embedding to a classifier containing one hidden layer.

3.3. Multi-Task Learning

Whereas single-task models lead benchmarks across translation, summarization and multi-label classification, multi-task models are proposed to capture multiple aspects of single tasks in order to combine them and exploit commonalities for increased performance. The objective of multi-task learning is to achieve transfer learning between multiple related tasks. Works in MTL primarily include text-based tasks with text inputs while introducing shared layers for capturing common shared information

[4] `http://eurovoc.europa.eu/drupal/`
[5] `https://eur-lex.europa.eu/homepage.html`

[68, 69, 70, 71, 72]. Kaiser et al. [4] introduced the MultiModel and extended the input capabilities to different domains through training specific input modality nets, thereby showing performance increases in translation, speech recognition and parsing tasks through joint training. To our knowledge, multi-task learning on deep neural networks has not been applied in the legal domain before.

4. Legal Datasets for Translation, Summarization and Classification

4.1. An Overview of Existing Corpora

In the following, an overview of existing legal corpora is given. Most of the depicted datasets are provided by entities of the European Union, while others were compiled by independent researchers working in the legal domain. Respective links to locations where the copora can be downloaded are attached to each section. Alongside this overview of legal corpora, a reference to the OPUS project [39] is expedient, which offers the possibility to freely download parallel datasets from various domains from a central point[1] including some of the subsequent mentioned corpora. All presented data collections are at least available in the English language. Accompanying tables show concentrated information on each corpus. A succinct overview of all datasets can be found in the Appendix A.

Proceedings of the European Parliament (Europarl)

The *Europarl corpus* (Europarl) [2] is a widely established collection of multilingual legal text from the European Union, which finds constant application in the NLP community for learning phrase representations [73], statistical machine translation [74] and more recent deep learning models [4, 5]. Published in 2005 by Philipp Koehn, the corpus provides parallel text for 20 languages. After a revision in 2012, the dataset comprises over 30 million sentences and fragments, like titles and exclamations, by now. The content comprises proceedings of the European Parliament scraped from its respective website[2] between the years 1996 and 2011. Thus, the sentences relate to discussions in political topics. Frequently, samples contain first-person narrative text expressing political opinions and positions. References at all, especially to codes of law are surprisingly rare. Since the text originates from spoken word the use of long connected sentences is generally omitted. This way, the Europarl makes a perfect fit for the legal translation task.

[1] `http://opus.nlpl.eu`
[2] `http://www.europarl.eu.int/`

Europarl	
Content	Proceedings of the European Parliament
Languages	Bulgarian, Czech, Danish, Dutch, English, Estonian, German, Greek, Finnish, French, Hungarian, Italian, Latvian, Lithuanian, Polish, Portuguese, Romanian, Slovak, Slovenian, Spanish, Swedish
Format	Moses/Giza++
Alignment	Sentence aligned
Size	30.11 million sentences
Timespan	1996 - 2011
Availability	Free
Download	`http://www.statmt.org/europarl/` `http://opus.nlpl.eu/Europarl.php`

Table 4.1.: Europarl corpus information

Directore-General for Translations - Translation Memory (DGT-TM)

With the *Directore-General for Translations Translation Memory* (DGT-TM) [38], an extensive translation memory consisting of parallel text in 24 languages is provided by the *Directore-General for Translations*[3] and *Joint Research Centre*[4] (JRC) of the European Union. At the time of this writing, there are over 65 million translation units[5] present in the corpus. An update to its data content is intended to happen on a yearly basis, thus serving as steadily growing resource for researchers in the NLP area. Texts of the DGT-TM are derived from legislative documents dealing with legal acts of the European Union[6]. Documents of the *Legislation* series of the *Official Journal* (OJ) are included, thence supply to the *Acquis Communautaire* and also constitute to the JRC-Acquis corpus 4.1. In comparison to other multilingual corpora, a superior quality of translations results from a fastidious revision process conducted by legal services and the Publications Office of the European Union[7]. Thereby, the major focus is on ensuring terminology consistency, which is additionally supervised by the public administrations of the EU member states. Despite the texts' natures and the assumption, that the corpus may only be useful for legal translation, an application in named entity recognition or sentiment analysis is eligible, as others have already proved the importance of such parallel corpora [75, 76].

[3] `https://ec.europa.eu/info/departments/translation`
[4] `https://ec.europa.eu/info/departments/joint-research-centre`
[5] Translation units are synonymous to sentences, additionally headers and titles are included
[6] e.g. texts of the EUR-Lex - `http://eur-lex.europa.eu`
[7] `http://ec.europa.eu/ipg/basics/management/day_to_day/opoce/index_en.htm`

DGT-TM	
Content	European Union's legislative documents
Languages	Bulgarian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Greek, Hungarian, Italian, Irish, Latvian, Lithuanian, Maltese, Polish, Portuguese, Romanian, Serbo-Croatian, Slovak, Slovenian, Spanish, Swedish
Format	TMX
Alignment	HunAlign
Size	65.49 million sentences
Timespan	2007 - 2018
Availability	Free
Download	`https://ec.europa.eu/jrc/en/language-technologies/dgt-translation-memory#download` `http://opus.nlpl.eu/DGT.php`

Table 4.2.: DGT-TM corpus information

Digital Corpus of the European Parliament (DCEP)

The *Digital Corpus of the European Parliament* (DCEP) [1] is yet another corpus published by the Joint Research Centre of the European Union in 2014. The DCEP states with over 1.5 million documents the largest single release of records by a European Union Commission. Enclosed alignment information (created with HunAlign sentence aligner [77] with same approach as for the JRC-Acquis 4.1) allows the generation of parallel text for over 250 language pairs (23 languages). Text of the DCEP spreads over diverse areas including press releases, session protocols, reports of the parliamentary committees and written questions. In order to obviate overlap with the Europarl corpus 4.1, protocols of speeches of the plenary are excluded from the DCEP.

DCEP	
Content	Agendas of plenary sessions, Parliamentary News, Press Releases, Motions for Resolutions, Plenary Sitting Protocols, Reports of the Parliamentary Committees, Rules of Procedure of the European Parliament, Final Texts of Plenary Votes, Written Questions
Languages	Bulgarian, Czech, Danish, Dutch, English, Estonian, German, Greek, Finnish, French, Hungarian, Italian, Irish, Lithuanian, Latvian, Maltese, Polish, Portuguese, Romanian, Slovak, Slovenian, Spanish, Swedish
Format	SGML and XML documents
Alignment	Enclosed HunAlign information
Size	1.5 million documents
Timespan	2001 - 2012
Availability	Free
Download	`https://wt-public.emm4u.eu/Resources/DCEP-2013/DCEP-Download-Page.html`

Table 4.3.: DCEP corpus information

European Union's Directorate General for Education and Culture - Translation Memory (EAC-TM)

The *European Union's Directorate General for Education and Culture Translation Memory* (EAC-TM) [78] is with just over 77000 translation units a tiny translation memory in comparison to other major legal corpora. The European Union's *Directorate General for Education and Culture* released the dataset in 2012. Despite its size, a variety of 26 languages is present. Electronic forms, such as report and application documents for decentralized actions of EAC's learning program, give substance to the corpus.

EAC-TM	
Content	Electronic forms for decentralized actions of EAC's learning program
Languages	Bulgarian, Croatian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Greek, Hungarian, Icelandic, Italian, Latvian, Lithuanian, Maltese, Norwegian, Polish, Portuguese, Romanian, Slovak, Slovenian, Spanish, Swedish, Turkish
Format	TMX
Alignment	HunAlign
Size	78613 sentences
Timespan	2012
Availability	Free
Download	`https://ec.europa.eu/jrc/en/language-technologies/eac-translation-memory`

Table 4.4.: EAC-TM corpus information

The EU Bookshop Corpus (EUbooks)

A large amount of parallel text can be found in the *EU Bookshop corpus* (EUbooks) [79]. This corpus contains publications in 24 languages from the European Unions's bookshop which is the online service and archive for various European institutions. The content spans over all aspects of the European Union's scope of duties, such as consumer rights, asylum, EU funding, transport, energy and EU law. While this corpus cannot be labeled as pure legal corpus, it contains texts which are nearby the legal domain. Moreover, the translations in the corpus are of high quality and subsequently best suited for training translation models.

EUbooks	
Content	Publications of european institutions in EU funding, consumer rights transport, energy, immigration and EU law
Languages	Bulgarian, Croatian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Greek, Hungarian, Irish, Italian, Latvian, Lithuanian, Maltese, Polish, Portuguese, Romanian, Slovak, Slovenian, Spanish, Swedish
Format	TMX and Moses/Giza++
Alignment	HunAlign
Size	172 million sentence fragments
Timespan	2014
Availability	Free
Download	`http://opus.nlpl.eu/EUbookshop.php`

Table 4.5.: EUbooks corpus information

Joint Research Centre - Acquis Communautaire (JRC-Acquis)

In 2006, the *Joint Research Centre* (JRC) of the European Union published the *JRC-Acquis corpus* [3]. A collection of legislative documents, retrieved from the European Union (EU) law applicable, the Acquis Communautaire [80], stating EU laws and policies which have to be implemented by each member state. Meanwhile, the corpus has been extended. The current version 3.0 contains roughly 21000 documents for each of the 22 available languages. All of the documents are fully annotated in XML according to the TEI guidelines [81]. The JRC also provides corresponding alignment information for the body paragraphs produced by two different aligning methods - Vanilla [82] and HunAlign [77]. This information allows the creation of parallel texts which are optimally suited for machine translation. Furthermore, a major part of the documents include manually assigned *EuroVoc* codes. These codes result from the *EuroVoc thesaurus*[8], a hierarchical classification structure introduced by the European Union. With over 6000 classes, documents are thoroughly allocated to multiple domains and subdomains such as agriculture, food, health, economy, information technology, law and politics. Hence, the EuroVoc annotations create valuable opportunities for automatic domain-specific terminology generation [83] and the training of multi-label document classifiers [60, 63].

[8] `http://eurovoc.europa.eu/`

JRC-Acquis	
Content	European Union's legislative documents / Acquis Communautaire
Languages	Bulgarian, Czech, Danish, Dutch, English, Estonian, German, Greek, Finnish, French, Hungarian, Italian, Lithuanian, Latvian, Maltese, Polish, Portuguese, Romanian, Slovak, Slovenian, Spanish, Swedish
Format	TEI-compliant XML annotated documents
Alignment	Vanilla and HunAlign
Size	463792 documents
Timespan	1958 - 2006
Availability	Free
Download	`https://ec.europa.eu/jrc/en/language-technologies/jrc-acquis` `http://opus.nlpl.eu/JRC-Acquis.php`

Table 4.6.: JRC-Acquis corpus information

Judgements of the House of Lords (The HOLJ Corpus)

The *HOLJ Corpus* [40] is a collection of judgments undertaken by the *House of Lords*. In contrast to other described corpora, the HOLJ corpus serves primarily as a source for summarization tasks and is therefore only available in the English language. In exchange, the corpus is extensively annotated. A manual annotation of the sentences was conducted in order to extract the rhetorical role of single expressions according to Teufel and Moens [84]. In addition, documents were automatically processed. Sentences got tokenized and POS-tagged with the LT TTT program [85], lemmatized according to [86] besides the extraction of named entities with the C&C named entity tagger [87].

The HOLJ Corpus	
Content	Judgments of the House of Lords
Languages	English
Format	XML annotated documents
Alignment	-
Size	188 documents
Timespan	2001 - 2003
Availability	Free
Download	`https://www.inf.ed.ac.uk/research/isdd/admin/package?download=84`

Table 4.7.: The HOLJ Corpus information

Proceedings of the Old Bailey (Old Bailey)

Exclusively in the English language, another corpus is at hand with the *Old Bailey* dataset [48]. The corpus contains proceedings of the historical *London Central Criminal Court*. All of the 1219 documents are XML annotated in-depth with various tags including offence descriptions, verdict descriptions, punishment descriptions, juror names, victim names, defendant names and location names. The historical origin of this corpus is reflected in its speech, however the share of full text has its limits.

Old Bailey	
Content	Proceedings of the London Central Criminal Court (Old Bailey)
Languages	English
Format	XML annotated documents
Alignment	-
Size	1219 documents
Timespan	1674 - 1834
Availability	Free
Download	`https://www.oldbaileyonline.org/static/Data.jsp`

Table 4.8.: Old Bailey corpus information

A Multilingual Corpus from the United Nations (MultiUN)

The *MultiUN corpus* (MultiUN) [88] comprises over 80 million sentences in Arabic, Chinese, English, French, German, Russian and Spanish. Derived from the Official Document System of the United Nations, about half of the corpus (German documents constitute 1%) covers three languages from outside Europe. All documents were cleaned from pictures, tables and figures. Subsequently, sentences were split accordingly with the NLTK toolkit [89] alongside appending tags for special phrases (currently only URLs and email addresses). After preparation, sentences of different languages were aligned with the HunAlign sentence aligner [77]. Supplementary, a common test set for the translation tasks is enclosed. The authors plan to release a new version of the corpus every half a year.

MultiUN	
Content	Official Documents of the United Nations
Languages	Arabic, Chinese, English, French, German, Russian, Spanish
Format	XML and Moses
Alignment	Sentence aligned
Size	80 million sentences
Timespan	2000 - 2009
Availability	Free
Download	`https://conferences.unite.un.org/uncorpus` `http://opus.nlpl.eu/MultiUN.php`

Table 4.9.: MultiUN corpus information

4.1.1. Text-to-text Input Formats

On the one hand, different models require different formats of training data, on the other hand, text corpora often exclusively exist in XML annotated form. These annotations vary from dataset to dataset and make it difficult to realize a standardized way of processing data. This includes the conversion into a suitable form, the combination of multiple datasets and the actual input to a model. Against that, modern deep learning frameworks provide their own tools to handle training data. Moreover, deep learning models are more frequently applied to raw data. This development often circumvents parts of forwarding the data through extensive processing pipelines to transform it to a task-dependent shape. Subsequently, we present different input formats for textual data and aspects to keep in mind while preparing data for translation, summarization and classification.

Translation Memory eXchange (TMX)

Naturally, aligned text is the preferred input format for translation tasks. The *Translation Memory eXchange* (TMX) specification serves this purpose and defines a database containing structured translation units. A translation unit consists of two segments, a sentence in the origin language and its respective translated sentence in the target language. Sentence pairs are segregated by XML tags from each other and establishing a clear view on the translation units. Each translation segment is additionally annotated with its language. The TMX format stipulates storing both sides of a translation unit into one file. On occasion, such a file can grow to respectable sizes rather quickly. Especially reappearing annotations contribute to this phenomenon. Even so, TMX is a widely-used format by providers of translation software and allows an easy reconstruction of the underlying source and target documents.

None of the major frameworks, including *Tensorflow* [90], Torch [91, 92], Theano [93], Caffe [94] or Deeplearning4J [95], used for deep learning natively supports the TMX

format or provides respective tools to easily handle this format. As a consequence, developers need to design their own tools or use a third-party XML parser to effectively import TMX. This circumstance hampers the ease in handling TMX and counteracts the movement towards raw data in deep learning.

Furthermore, TMX is specifically tailored for the structured storage and transmission of translation data. Other text-to-text problems, such as summarization, are not taken into account. Thus, TMX cannot be reliably used across different tasks, whereas the course of action is similar to translation. As part of this work, a simpler format was required with regard to equally fulfill the need of translation, summarization and classification data requirements.

Listing 4.1: Excerpt of a TMX file for the German-to-English language pair from the MultiUN corpus

```
<?xml version="1.0" encoding="UTF-8" ?>
<tmx version="1.4">
<header creationdate="Tue␣Feb␣19␣01:31:42␣2013"
        srclang="de"
        adminlang="de"
        o-tmf="unknown"
        segtype="sentence"
        creationtool="Uplug"
        creationtoolversion="unknown"
        datatype="PlainText" />
  <body>
    <tu>
      <tuv xml:lang="de"><seg>Internationale Zusammenarbeit und
          Koordinierung für die Wiederherstellung der Gesundheit der Bev
          ölkerung, die Sanierung der Umwelt und die wirtschaftliche
          Entwicklung der Region Semipalatinsk in Kasachstan</seg></tuv>
      <tuv xml:lang="en"><seg>International cooperation and coordination for
          the human and ecological rehabilitation and economic development of
          the Semipalatinsk region of Kazakhstan</seg></tuv>
    </tu>
    <tu>
      <tuv xml:lang="de"><seg>unter Hinweis auf ihre Resolutionen 52/169 M
          vom 16. Dezember 1997, 53/1 H vom 16. November 1998, 55/44 vom 27.
           November 2000 und 57/101 vom 25. November 2002,</seg></tuv>
      <tuv xml:lang="en"><seg>Recalling its resolutions 52/169 M of 16
          December 1997, 53/1 H of 16 November 1998, 55/44 of 27 November
          2000 and 57/101 of 25 November 2002,</seg></tuv>
    </tu>
    ...
  </body>
</tmx>
```

Moses

A much simpler format than TMX got established with the Moses statistical machine translation toolkit [96] by dividing the source translations and the target translations into two separate files. Hereby, the translations do not contain any annotations and are aligned line by line. Obviously, files typically come in pairs to facilitate training on translation tasks. Moreover, subjoining translation files of additional languages may be possible as long as the alignment complies. This is often seen in test set compilations,

in which content variations throughout multiple language pairs should be mitigated. Though, the attachment of files with alignment information is common to allow the reconstruction of the original documents as well as the adjustment between all language pairs.

The simplicity of this format allows an ease in application by major machine learning frameworks. Since annotations are avoided, there is no need for a specific parser. Likewise, reading associated files, can be achieved pretty quickly with several programming languages, especially with Python, the most used language among the popular machine learning frameworks.

In contrast to TMX, Moses can just as well be used to format the input data for other text-to-text problems, e.g. stating full texts and summaries for summarization or paragraphs and classes for classification. This flexibility in combination with the simplicity makes the Moses format very useful to text processing and chimes in with the demand for raw data by deep learning applications. These properties made us to choose the Moses format for this work.

Listing 4.2: Excerpt of a Moses file for the source language English from the MultiUN corpus

```
We reaffirm our faith in the United ...
Emphasizing the importance of ...
The General Assembly
68th plenary meeting 22 December 2005
This is our shared responsibility ...
Some countries will implement ...
Domestic resource mobilization
To this end, we therefore resolve ...
These initiatives could include ...
A universal, rule−based, open, ...
...
```

Listing 4.3: Excerpt of a Moses file for the target language German from the MultiUN corpus

```
Wir bekräftigen unseren Glauben an ...
Betonend, wie wichtig die ...
Die Generalversammlung
68. Plenarsitzung 22. Dezember 2005
Dies ist unsere gemeinsame ...
Einige Länder werden die ...
Mobilisierung einheimischer Ressourcen
Zu diesem Zweck beschließen wir ...
Derartige Initiativen könnten ...
Ein universales, üregelgesttztes, ...
...
```

4.2. Ready-to-use Legal Corpora for Translation, Summarization and Classification

The first contribution of this work is the preparation of multiple corpora for legal translation, text summarization and document classification. Despite the existence of numerous legal corpora, especially for the translation task, no corpus except the Europarl originally comes in the Moses format. However, the Moses format is the preferred shape for conducting sequence-to-sequence tasks on textual data with models based on deep neural networks. Moreover, test sets have to be extracted. This leads to researchers creating deviant test sets which likely impact a fair comparison of different approaches to solve a task. Hence, we provide ready-to-use legal copora with extracted test sets in the Moses format for translation, summarization and classification. In chapter 6, we are going to use these datasets for our experiments with the MultiModel. The resulting corpora are available at *MediaTUM* (see table 4.15). The programming scripts used to process the corpora are also publicly available[9].

4.2.1. Three Corpora for Legal Translation

Despite compiling corpora from new resources for legal translation, we fall back on the Europarl, the DCEP and the JRC-Acquis which already contain massive sources of parallel text. These specific corpora have been selected, because each corpus provides a different legal discourse type. Furthermore, the corpora do exist in over 20 languages and have a pleasant size to be considered for deep learning applications. Beforehand, we chose a subset of 7 languages to be included in the processed corpora. The chosen languages are 3 Romanic languages (French/FR, Italian/IT, Spanish/ES), 3 Germanic languages (German/DE, English/EN, Swedish/SV) and 1 Slavic language (Czech/CS). These 7 languages lead to 21 language pairs, for which we created according training and test sets. It is important to mention that the amount of samples across all Czech tasks is nearly 50% smaller than for the other tasks, since less Czech documents are available in the original corpora (see table 4.10). The processing steps differ between the Europarl, DCEP and JRC-Acquis. Therefore, we split this section into three parts.

Legal DCEP (legal-dcep)

The DCEP needed extensive preprocessing, since the corpus has not been converted to Moses format yet. The dataset provided by the Joint Research Centre of the European Union contains all documents in XML and SGML format. The documents need to be flattened in order to be fed into sequence-to-sequence translation models. The JRC provides scripts to convert the DCEP. However, these scripts are not capable to shape the corpus into Moses format. Therefore, we deployed our own script for this

[9] `https://github.com/cgebe/m-thesis-scripts`

transformation. Enclosed alignment information files contain line numbers, but are separated document-wise for each language pair. As follows, instead of selecting specific line numbers for exclusion, we were forced to produce test set alignment information containing randomly selected document identification numbers in order to ensure the same content across test sets. We used the English-to-French data to pick 2% of the documents which approximately translates to 2% of the sentence pairs. The resulting alignment information was utilized to filter out respective translation units for all 21 language pairs. After creating the test sets, we removed sentence pairs with less than 25 characters, due to impure short sentence fragments mostly containing only special characters. For the sake of integrity, we did not filter out these short fragments from the training sets.

Legal Europarl (legal-europarl)

The Europarl corpus is on hand in aligned form (Europarl-v7), therefore additional processing to produce parallel text in the Moses format is not required. Moreover, an alignment information file is attached to every two files representing a language pair. These files contain document ids with line numbers derived from the original documents. Therefore, the alignment information can be used to extract test set samples for each language pair. Again, we randomly selected 2% of the lines from the English-to-French alignment information file to produce a test set alignment information file for maximum coverage across all language pairs. Based on this file, we filter out the according samples for all the other pairs by intersecting the selected line numbers with their alignment information. This leaves us with test sets covering the same content. Nonetheless, test sets differ in size due to varying amounts of source documents. E.g., Czech pairs are about 1/4 of the size of other pairs.

	legal-europarl		legal-dcep		legal-jrc-acquis		Combined	
Task	Train	Test	Train	Test	Train	Test	Train	Test
cs-de	554785	12877	3322863	26365	956206	7926	4833854	47168
cs-en	632331	13475	3429669	26023	954139	7731	5016139	47229
cs-es	605198	13293	3331565	24771	965210	8003	4901973	46067
cs-fr	614135	12797	3353646	27185	961109	7979	4928890	47961
cs-it	593176	12533	3427214	25216	956954	7939	4977344	45688
cs-sv	617190	13088	3353962	24759	912310	7447	4883462	45294
de-en	1920519	39310	5389867	47179	1227338	9800	8537724	96289
de-es	1848928	38030	5244580	46939	1234976	10129	8328484	95098
de-fr	1904020	37733	5353860	50266	1239724	10100	8497604	98099
de-it	1794869	37183	5399213	47908	1230221	10098	8424303	95189
de-sv	1803663	37445	5223641	45001	1150149	9416	8177453	91862
en-es	1968689	39069	5782727	50650	1231178	9916	8982594	99635
en-fr	2011292	38370	5730964	52431	1237570	9916	8979826	100717
en-it	1907616	37237	5617352	45408	1222257	9874	8747225	92519
en-sv	1854436	37007	5684684	48163	1144536	9300	8683656	94470
es-fr	1944351	37359	5507250	50581	1244162	10234	8695763	98174
es-it	1843115	36557	5458678	46341	1242123	10229	8543916	93127
es-sv	1789877	35696	5539768	46588	1155662	9382	8485307	91666
fr-it	1906101	36276	5558798	49214	1234846	10196	8699745	95686
fr-sv	1842905	36220	5321630	45969	1157695	9423	8322230	91612
it-sv	1730666	34133	5402000	43877	1155158	9354	8287824	87364
Total	**31687862**	**635688**	**102433931**	**870834**	**23813523**	**194392**	**157935316**	**1700914**

Table 4.10.: Number of translation units in training and test sets of the legal translation corpora (legal-dcep, legal-europarl, legal-jrc-acquis)

Legal JRC-Acquis (legal-jrc-acquis)

The original JRC-Acquis corpus contains documents in XML format which are sorted by language and year. The body of the documents constitute the important part for the generation of a translation corpus. Each paragraph in the body of a document is annotated with a paragraph number which is used for producing parallel text. The enclosed alignment information files comprise the source and target paragraph numbers document-wise for each language pair. This information allowed us to align the paragraphs accordingly and produce parallel text for 21 language pairs. Beforehand, we randomly selected 2% of the documents based on the document id for the separation of the test set samples. While producing the aligned text, we simply filtered out paragraphs of matching documents in order to add them to the test set. We applied the identical cleaning process of the DCEP test set to the JRC-Acquis. We

removed all translation units where either of the sentences was less than 25 character long. Analogously to the DCEP preparation, we did not remove these samples from the training sets.

4.2.2. A Corpus for Summarizing Legislative Texts

Despite the HOLJ corpus, there are no preprocessed alternative datasets which can be used for training legal summarization. Due to legal datasets being sparse, we draw data from existing legal corpora and bring it into a format so that it can be used for summarization in multi-task as well as single-task deep learning.

JRC-Acquis Summarization Corpus (legal-jrc-acquis-summmarize)

For the preparation of a summarization corpus, we process the JRC-Acquis. Due to its rich annotations, the JRC-Acquis places an excellent foundation at the disposal for the creation of such a corpus. Each JRC-Acquis document contains a title element holding a short description of the document body. This summary usually consists of up to three sentences representing the semantic core of a document. Congruently to the translation corpora, the same seven languages (CS, DE, EN, ES, FR, IT, SV) were selected for the construction of the corpus. However, in contrast to translation and the resulting 21 language pairs, each of the seven languages corresponds to one task within legal summarization. We extract the document body and the short description of each document for these seven languages. For a big part of the documents the body needed additional processing. Documents frequently repeated the title content at the beginning of the document body. Therefore, an adjustment was undertaken to remove the respective body paragraphs whenever they occurred multiple times in order not to tamper the mapping between full text and summary. Additionally, full

legal-jrc-acquis-summarize		
Task	Train	Test
cs	17956	264
de	22707	327
en	22448	328
es	22751	327
fr	22586	326
it	22371	322
sv	19255	265
Total	**150074**	**2159**

Table 4.11.: Number of samples in training and test sets of the legal summarization corpus (legal-jrc-acquis-summarize)

text-summary pairs with more than 15k characters of full text were excluded from the corpus which lead to a reduction by less than 1% of the examples. The corpus is produced in the Moses format. This means, each sample which consists of the full text as input and the summary as label is aligned line by line across two files. As mentioned previously, the Moses format offers a comfortable way for feeding the data into sequence-to-sequence models without being limited only to the storage and transmission of translation corpora.

4.2.3. Corpora for the Classification of Legal Documents (legal-jrc-acquis-label)

The classification of legal documents is one of the most prominent task executed on legal text. For this reason, we provide several corpora for promoting experiments in this task family.

JRC-Acquis Multi-label Classification Corpus (legal-jrc-acquis-label)

A valuable source for the creation of a legal classification corpus is given with JRC-Acquis. The JRC-Acquis documents include EuroVoc thesaurus annotations which are perfectly suited to setup a multi-label classification task. The amount of classes per document usually ranges between one and seven classes. The classes are annotated as numbers and located in the header element of each document. A part of older JRC-Acquis do not contain these annotations and are therefore neglected in the resulting dataset. Documents in seven languages (CS, DE, EN, ES, FR, IT, SV) were chosen to be processed. We extract the body paragraphs and the respective EuroVoc classes from the header of each document. Subsequently, extracted examples are transferred into Moses format. Body paragraphs are stripped of new lines and squeezed onto one line,

legal-jrc-acquis-label		
Task	Train	Test
cs	12571	253
de	14153	295
en	14391	302
es	14065	296
fr	14147	297
it	14086	293
sv	11561	236
Total	**94974**	**1972**

Table 4.12.: Number of samples in training and test sets of the legal labeling corpus (legal-jrc-acquis-label)

whereas class numbers are concatenated and delimited by space. While producing all examples, we reserve 2% for the test set. Therefore, we randomly selected documents which are filtered out across all languages.

Legal GCD - German Court Decisions Corpus (legal-gcd)

Finally, we compiled a new corpus which we call *Legal GCD* (legal-gcd). The basis of this corpus are court decisions from the seven highest courts in Germany. This includes the *Bundesverfassungsgericht* (BVerfG), *Bundesgerichtshof* (BGH), *Bundesverwaltungsgericht* (BVerwG), *Bundesfinanzhof* (BFH), *Bundesarbeitsgericht* (BAG), *Bundessozialgericht* (BSG) and *Bundespatentgericht* (BPatG). Court decisions of these courts are made publicly available through a search engine on the Internet[10]. We scraped the XML version of each available document on the 30.01.2018. This resulted in 42683 documents. See table 4.13 for more information. Documents can be categorized into the following three document types besides few additional types which only occur in connection with single courts:

Resolution A resolution is a form of a court decision which often emanates from a civil process without court hearing.

Verdict A verdict is a form of a court decision which emanates from a process with court hearing.

EuGH Draft These are drafts which are forwarded to the *court of the European Union* (EuGH)

The type determines which document sections are filled. Nonetheless, documents specify empty tags for sections they do not provide. This facilitates consistent processing

[10] `http://www.rechtsprechung-im-internet.de`

legal-gcd					
Court	Resolutions	Verdicts	EuGH Drafts	Other	Total
BAG	588	3972	21	-	4581
BFH	3607	3838	60	-	7505
BGH	8466	5874	99	6	14445
BPatG	4399	302	10	-	4711
BSG	1314	2000	2	-	3316
BVerfG	2168	47	2	490	2707
BVerwG	3517	1860	37	4	5418
Total	**24059**	**17893**	**231**	**500**	**42683**

Table 4.13.: Number of documents of the legal corpus (legal-gcd)

of the documents. Each document provides a header section for general information, a tenor for summarizing the outcome, facts stating the basis of a case, decision reasons and a footer. See figure 4.4 for a example document. In addition to the XML documents, pictures relevant to court decisions are enclosed. For example, BPatG decisions which are mainly patent resolutions usually come with pictures related to the processed patent.

German Court Decisions Classification Corpora (legal-gcd-court & legal-gcd-verdict)

We processed the legal-gcd further in order to construct two classification tasks which are trained with legal-gcd documents. The first classification task includes assigning documents to the court they belong to. More specifically, we extract the 'facts' section of each verdict document and align it with the court to which the verdict belongs to. This results in 15884 verdict facts being aligned with their originating court as training set. 2% of the verdicts were reserved as test set. Again, we use the Moses format to store the samples in an accessible and easy way. We call this corpus *legal-gcd-court*.

The second ready-to-use corpus which we produced from the legal-gcd is the *legal-gcd-verdict*. For this dataset, we extracted the facts of each verdict document with the outcome of the verdict. The outcome of a verdict is inferred by keywords in the tenor of the verdict which states a short summary of each incident. The outcome can be one of two classes, positive or negative. A positive outcome usually stands for a successful verdict revision, whereas a negative outcome reflects the denial of a revision. Verdict documents with more complex outcomes and missing keywords were neglected. The resulting dataset contains 11483 examples with additional 228 which were reserved for the test set.

	legal-gcd-court		legal-gcd-verdict	
Task	Train	Test	Train	Test
de	15884	318	11483	228

Table 4.14.: Number of samples in training and test sets of the legal classification corpora (legal-gcd-court & legal-gcd-verdict)

Listing 4.4: Excerpt of a document from the Legal GCD

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dokument
  SYSTEM "http://www.rechtsprechung-im-internet.de/dtd/v1/rii-dok.dtd">
<dokument>
   <doknr>JURE100055435</doknr>
   <ecli/>
   <gertyp>BGH</gertyp>
   <gerort/>
   <spruchkoerper>Senat für Anwaltssachen</spruchkoerper>
   <entsch-datum>20100107</entsch-datum>
   <aktenzeichen>AnwZ (B) 79/09</aktenzeichen>
   <doktyp>Beschluss</doktyp>
   <norm>§ 14 Abs 2 Nr 7 BRAO, §259 Abs 1 S 2 InsO, §291 Abs 1 InsO</norm>
   <vorinstanz>vorgehend OLG Hamm, 24. April 2009, Az: 1 AGH 11/09, Beschluss<br/>
   </vorinstanz>
   <region>
      <abk>DEU</abk>
      <long>Bundesrepublik Deutschland</long>
   </region>
   <mitwirkung/>
   <titelzeile>
      <dl class="RspDL">
         <dt/>
         <dd>
            <p>Anwaltliches Berufsrecht: Widerruf der Anwaltszulassung bei Eröffnung des Insolvenzverfahrens bei
                  Kanzleifreigabe und Beantragung der Restschuldbefreiung</p>
         </dd>
      </dl>
   </titelzeile>
   <leitsatz/>
   <tenor>
      <div>
         <dl class="RspDL">
            <dt/>
            <dd>
               <p>Die sofortige Beschwerde der Antragstellerin gegen den Beschluss des 1. Senats des Anwaltsgerichtshofs
                     des Landes Nordrhein-Westfalen vom 24. April 2009 wird zurückgewiesen.</p>
            </dd>
         </dl>
         ...
      </div>
   </tenor>
   <tatbestand/>
   <entscheidungsgruende/>
   <gruende>
      <div>
         ...
         <dl class="RspDL">
            <dt>
               <a name="rd_2">2</a>
            </dt>
            <dd>
               <p>Das nach §215 Abs. 3 BRAO i.V.m. §42 Abs. 1 Nr. 2, Abs. 4 BRAO a. F. zulässige Rechtsmittel bleibt ohne
                     Erfolg.</p>
            </dd>
         </dl>
         ...
      </div>
   </gruende>
   <abwmeinung/>
   <sonstlt/>
   <identifier>http://www.rechtsprechung-im-internet.de/jportal/?quelle=jlink&amp;docid=JURE100055435&amp;psml
         =bsjrsprod.psml&amp;max=true</identifier>
   <coverage>Deutschland</coverage>
   <language>deutsch</language>
   <publisher>BMJV</publisher>
   <accessRights>public</accessRights>
</dokument>
```

Corpus	Type	Link
legal-dcep	Translation	https://mediatum.ub.tum.de/1446648
legal-europarl	Translation	https://mediatum.ub.tum.de/1446650
legal-jrc-acquis	Translation	https://mediatum.ub.tum.de/1446655
legal-jrc-acquis-summarize	Summarization	https://mediatum.ub.tum.de/1446654
legal-jrc-acquis-label	Classification	https://mediatum.ub.tum.de/1446653
legal-gcd, legal-gcd-court & legal-gcd-verdict	Classification	https://mediatum.ub.tum.de/1446651

Table 4.15.: Links to MediaTUM for the download of the legal corpora

5. The Integration of Legal Corpora into the Multi-Task State-of-the-art Model

In order to train translation, summarization and classification on the newly created datasets, a model specifically able to perform multi-task deep learning on textual data needs to be applied. Within this work, we fall back to the state-of-the-art model for multi-task deep learning called *MultiModel*, proposed by Kaiser et al. [4] in June, 2017. In the following, we briefly present the MultiModel and describe our integration into the MultiModel via Tensor2Tensor [17].

5.1. The MultiModel - Everything Under One Roof

The MultiModel has been proposed to create a unified deep learning model which is able to solve tasks across multiple areas of neural network based research. This way, tuning a network for specific problems related to computer vision, speech recognition or natural language processing would become mitigated or completely obsolete. While the differentiation between domains primarily equates to differences in the areas of application as part of the MultiModel work, we focus to a greater extent on the content of the corpora used for training the tasks. Especially since, legal translation, summarization and text classification make only use of one of the input components of the MultiModel which is responsible for language related tasks. No changes have been made to the architecture of the MultiModel. Henceforth, we concisely depict its architecture and highlight the parts which played a key role for the integration of legal translation, summarization and text classification.

5.1.1. Architecture

The MultiModel consists of four essential parts facilitating multi-task learning across multiple different areas of application. The architecture is based on a fully convolutional sequence-to-sequence approach which includes three actors (encoder, decoder, mixer) similarly used by ByteNet [97] and WaveNet [98]. For the purpose of multi-task learning, the MultiModel uses so-called modality nets besides specially designed and adjusted computional blocks in its architecture. Below, all four parts are briefly presented. For more details, we refer to the MultiModel article [4].

Modality Nets Until now, there are four modality nets (language, image, audio, categorical) available in the MultiModel. A modality net is attached to the front and back of the MultiModel to deal with different inputs and outputs. The language, image and audio modality nets are responsible for the conversion of textual, image and audio input data into a variable-size joint representation which is fed to the encoder. On the other side, the language and categorical modality nets are used to transfer the variable size joint output of the decoder into the expected output format. Different tasks with the same input or output format share a modality net in order to promote generalization and allow the quick addition of further tasks.

Encoder The encoder takes the unified representation of an input modality net and processes it to produce encoded inputs. The encoder contains six custom-built convolutional blocks with one mixture-of-expert layer [99] in between. The contained convolutional operation is derived from the Xception architecture [100] and previous work by Kaiser et. al. [101].

Decoder The decoder takes the encoded inputs from the encoder and encoded outputs from the mixer to generate variable size decoded outputs for an output modality. The decoder consists of four convolutional-attention blocks with one mixture-of-expert layer in the middle. It is important that a command token is passed to the decoder at the beginning of each decoding run. This way the decoder learns an embedding for each problem allowing it to produce decoded outputs of different tasks for the same modality net.

I/O Mixer The mixer takes the encoded inputs from the encoder and unified outputs from the modality nets (from previous positions) to produce encoded outputs which are fed back into the decoder. The mixer comprises two convolutional blocks and one attention block. Apparently, the decoder and mixer pursue an autoregressive scheme by taking the encoded outputs from previous steps into account. This allows the MultiModel to learn long-term dependencies via large receptive fields in the convoluional blocks on inputs as well as former outputs.

Language Modality Net

The language modality net plays a key role in our work, since translation, text summarization and document classification are dealing with textual data. Therefore, we do only use this specific modality net for the conversion of the input text into the variable length internal representation fed to the encoder and mixer. This modality net tokenizes text according to the method from [102] using a subword vocabulary of 32k subwords. Then, the token sequence is mapped with a learned embedding to the size of the subsequent input layer. As output modality, the language modality takes

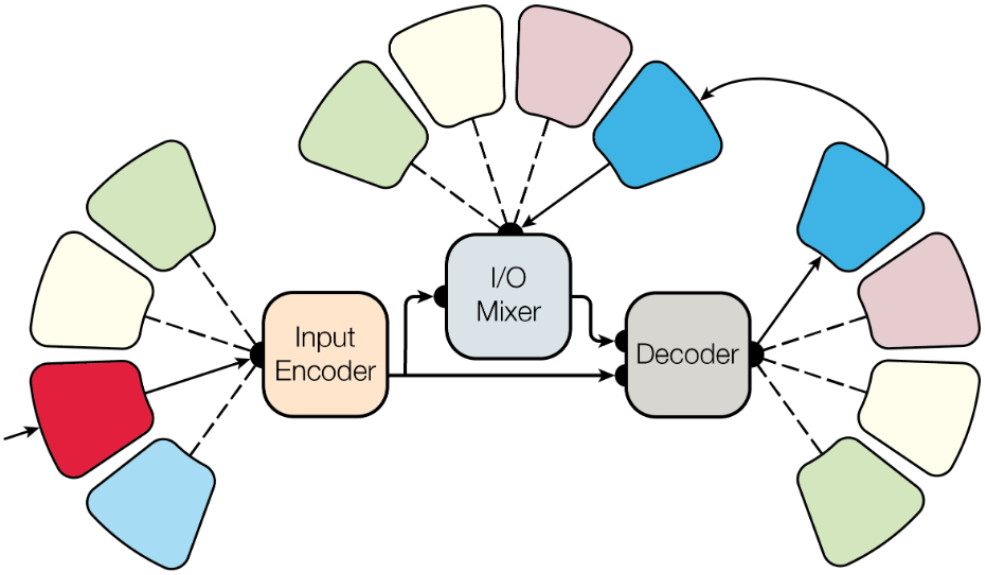


Figure 5.1.: MultiModel component overview [4]

the decoded outputs and produces text from the subword vocabulary according to a probability distribution provided by a *softmax* layer.

5.2. Implementation

This work is strongly building upon the MultiModel which allows the application of multi-task deep learning on different text problems. Fortunately, an implementation of the MultiModel was published[1] and exists as part of the Tensor2Tensor library [17]. By using this library, we are able to integrate our corpora respective to legal translation, text summarization and document classification with so-called data generators. These building blocks are responsible for defining and training custom problems within Tensor2Tensor.

5.2.1. Tensor2Tensor

Tensor2Tensor (T2T) is a library for deep learning models and datasets. It uses Tensorflow [90] under the hood and provides tools to rapidly prototype custom problems which can be integrated into T2T. T2T offers the possibility to easily apply multiple models for defined problems to make comparisons amongst architectures. There are

[1] `https://github.com/tensorflow/tensor2tensor/tree/v1.6.6/tensor2tensor/models/research`

over 10 different base models available. Models also come in numerous variants, including *hyperparameter* sets, which can be customized to fit special demands. At this point in time, the latest version of T2T is 1.6.5. This version does not contain the MultiModel, essential to this work. The support for multi-task learning with the MultiModel was dropped in version 1.4 and it got removed in version 1.5. Therefore, we switch to an older version still supporting multi-task learning with the MultiModel. The only version which fitted our requirements and worked flawlessly for text-to-text tasks is version 1.2.6[2]. This is the version we use to implement the data generators for the legal problems.

Mode of Operation

The operation with Tensor2Tensor is divided into three steps.

1. Data Generation
   Before training a model, the raw data (train and test data) which is used for a problem, needs to be transformed into a format so that it can be directly read by the model classes. Inside the data generator, it must be defined how the raw data is aligned to input-label pairs that state the examples for the training. The aligned pairs are shuffled and saved into *TFRecord* files. In addition, the vocabulary for the language modality net is build based upon the input and output data.

2. Training
   The content of the TFRecord files serves directly as input to the training process. Tensor2Tensor picks batches of examples from the TFRecord file and feeds them into the model. The graph parameters are then updated by the optimization process of *Tensorflow*. During the training process, checkpoints for the model are saved which contain all parameter values for the present training state. Based on these model checkpoints, Tensor2Tensor is capable to evaluate the model periodically to report the training process on a subset of the test set data. The training is usually stopped, as soon as the model converges towards a value on the selected evaluation metrics.

3. Decoding
   After completing the training process, it is possible to decode from the model either interactively or by providing a file. Now, the whole test set is given into the model to receive decodes for all unseen samples.

The implementation work as part of this work focuses on step 1, the data generation.

[2] `https://github.com/tensorflow/tensor2tensor/releases/tag/v1.2.6`

Data Generators

The purpose of a data generator is the generation of TFRecord files which contain a uniform representation of the training data. The content of the TFRecord files is directly fed into the model at training time. A generator contains problem definitions which state the surrounding conditions for experiments. By inheriting the Text2TextProblem class, a problem definition for a text-to-text problem includes the vocabulary size for the subword vocabulary, input/output space ids for embedding, metrics used for evaluation through the training process and the actual generator yielding the input-label pairs for training. The generator function dynamically downloads an associated corpora and subsequently extracts the contained examples line by line with appropriate Python tools. This circumstance is the main reason why the Moses format fits exceptionally well for providing corpora to text-to-text tasks. At this juncture, sequence encoders provided by T2T are deployed to encode text accordingly. We implement a data generator for each task family (translation, summarization, multi-labeling, multi-class classification). An example generator can be found in Appendix B. The source code for all generators is publicly available[3]. The following list depicts all legal problems defined in the newly created data generators. The problems are created analogously to the legal compiled datasets in chapter 4. The translation problems are trained jointly on all three translation corpora (legal-dcep, legal-europarl, legal-jrc-acquis). These can be reverted as well, however we limit our experiments to the originally defined pairs.

Translation (legal-dcep, legal-europarl, legal-jrc-acquis)
: Czech-German, Czech-English, Czech-French, Czech-Italian, Czech-Spanish, Czech-Swedish, German-English, German-French, German-Italian, German-Spanish, German-Swedish, English-French, English-Italian, English-Spanish, English-Swedish, Spanish-French, Spanish-Italian, Spanish-Swedish, French-Italian, French-Swedish, Italian-Swedish

Summarization (legal-jrc-acquis-sum)
: Czech, German, English, Spanish, French, Italian, Swedish

Multi-Label Classificiation (legal-jrc-acquis-label)
: Czech, German, English, Spanish, French, Italian, Swedish

Multi-Class & Binary Classification (legal-gcd-court, legal-gcd-verdict)
: German-Court, German-Verdict

Multi-Task Learning with Tensor2Tensor

A multi-task learning problem is not defined explicitly as a multi-task problem in T2T. After the definition of each problem as being single-task, it is possible to join the

[3] `https://github.com/cgebe/tensor2tensor`

problems before training. While conducting multi-task learning, T2T trains selected problems by concurrently feeding in data from the according TFRecord files generated for each separate problem. This allows maximum flexibility in joining different problems.

Language Output Modality for Classification

Indeed, we are using the language output modality throughout all problems. This includes the classification problems. The categorical output modality did not work properly in conjunction with the MultiModel, due to a bug in the used version. Therefore, it was necessary to fall back to encoding the classification output class as a token sequence. For the multi-label classification task, multiple classes are sorted in ascending order and separated by space. Single class classification problems enclose the class as a single word.

6. Experiments

Taking the newly created corpora and the integrated data generators, it is now possible to conduct a handful of experiments in regard to multi-task deep learning in the legal domain. As part of our work we do not only evaluate the MultiModel, but also compare it with the Transformer model. Both being state-of-the-art, a comparison on legal tasks facilitates new insights and helps to induce answers to our research questions. Before going into detail with the experiments, it is necessary to describe the setup we trained the models in.

6.1. Experimental Setup

6.1.1. Hardware

We used multiple different machines to train the models. The biggest reason for this approach is the time consuming training of each model besides model-dependent resource demands. The main differences between the machines are the associated GPUs. As mentioned before, Tensor2Tensor uses Tensorflow as underlying library to build and execute the computational graphs of the models. Tensorflow can use available GPUs to greatly speed up the training process which is indispensable when working with models and datasets of large sizes. Tensor2Tensor is capable to scale variably across multiple GPUs. Hence, changes to the data generators or other parts were not needed. The following table shows the specifications of the three machines we used to train the models.

		Machine 1	Machine 2	Machine 3 (DGX-1)
GPUs		4x GTX 1080 TI	4x Tesla K80	8x Tesla V100
Cores		~14k	~10k	~41k
Memory		4x 11GB	4x 12GB	8x 16GB
Training Steps	Translation	500k	500k	250k
	Summarization	100k	100k	50k
	Classification	100k	100k	50k
Training Time (dependent on 6.1.2)	Single-Task	25.2 s/100 steps	86.2 s/100 steps	86.4 s/100 steps
	Multi-Task (5 Tasks)	51.8 s/100 steps	-	155.5 s/100 steps

Table 6.1.: Machines used to train the models

6.1.2. Hyperparameters

We train the MultiModel with two different sets of hyperparameters in regard to research question number 4. The Transformer model is exclusively trained in the base configuration. Despite the hidden size, filter size and batch size, there are no changes between the light and base version of the MultiModel (see table 6.2). The complete set of hyperparameters can be found inside the model classes in the code repository[1].

	MultiModel Light (MM-L)	MultiModel Base (MM-B)	Transformer Base (TF-B)
Hidden Size	128	512	512
Filter Size	1024	2048	2048
Batch Size	1024	2048	2048
Total Parameters	~61m	~660m	~51m

Table 6.2.: Model hyperparameter sets

If not stated otherwise, the Multimodel Light was trained on machine 1, the Transformer Base on machine 2 and the MultiModel Base on machine 3. Additionally, if not explicitly mentioned, the training step count on machine 1 and machine 2 was set twice as high than on machine 3, due to the double amount of available GPUs on machine 3. When training on multiple GPUs, Tensor2Tensor mirrors the model on each GPU and feeds a different batch respectively. Finally, the computed gradients of all GPUs are merged to obtain a mean for the update. Therefore, the step count is adapted to the numbers of GPUs in order to pass trough the same amount of examples on each model. Conseqeuently, models were always trained on all GPUs of their respective machines to utilize the full available computing power while maintaining a fair environment for comparisons through adjusting the step count.

6.1.3. Metrics

We report our results with common task-dependent metrics. Translation results include BLEU [49] as well as CHRF [103] scores. The BLEU score measures the precision in distinct n-grams overlaps (1-gram to 4-gram) between the automatic translation and the reference translation. The BLEU metric is predestined to correlate well with human judgment and alleviates automatic evaluation of translation systems.

$$BLEU = min(1, \frac{hypothesis\_length}{reference\_length})(\prod_{i=1}^{4} precision_i)^{\frac{1}{4}}$$

[1] https://github.com/cgebe/tensor2tensor/tree/thesis/tensor2tensor/models

Additionally, we use the CHRF metric to project a second view onto the translation results. CHRF combines precision and recall of character n-grams to build an F-score based on character n-grams.

$$CHRF_{\beta} = (1 + \beta^2) \frac{chrP \cdot chrR}{\beta^2 \cdot chrP + chrR}$$

- *chrP*: percentage of character n-grams in the hypothesis which have a counterpart in the reference
- *chrR*: percentage of character n-grams in the reference which are also present in the hypothesis

In our evaluations, the character n-gram limit is set to 6 and we use a common beta value of 3 which is responsible for assigning 3-times more importance to recall than to precision. The tool used to apply the metrics onto the model decodes was *sacreBLEU* which is part of the Sockeye toolkit [104]. We use the ROUGE metric [47] to evaluate the summarization results, since it is commonly used by researchers to compare summarization performance. We focus on 1-grams, 2-grams and longest common subsequences, namely ROUGE-1, ROUGE-2 and ROUGE-L.

$$ROUGE_N = \frac{\sum_{S \in reference\_summaries} \sum_{gram_n \in S} count\_match(gram_n)}{\sum_{S \in reference\_summaries} \sum_{gram_n \in S} count(gram_n)}$$

The script used to apply the ROUGE metric was a Python script which binds to the ROUGE evaluation package of the Berkeley Document Summarizer [105]. For multi-label classification, we report accuracy, precision, recall and the percentage of at least one (At least 1) occurrence of a correct class computed through a custom written Python script[2].

6.2. Single-Task Training

To create a baseline for the multi-task investigations, we train the MultiModel Light, MultiModel Base and the Transformer Base on single tasks. This way, it was possible to produce reference results for the single task scenarios in order to confine the impact of multi-task training combinations. Due to strict time-constraints, numerous single-task training runs on the MultiModel Base and Transformer Base were dropped in favor of extensive multi-task experiments. In the following sections, we present single-task results and following insights of the different models.

[2] `https://github.com/cgebe/m-thesis-scripts`

6.2.1. Translation

Each single task training run included 500k training steps. The german-to-english translation task was the only task that could be trained on all models. We evaluate trained translation models on all 3 test sets provided by each corpus (legal-dcep, legal-europarl, legal-jrc-acquis).

The Transformer Base yields a BLEU score of 37.34 for the legal-europarl corpus and merely beats the MultiModel Base with 37.15. Against that, the MultiModel Base outperforms the Transformer Base on the legal-dcep and legal-jrc-acquis by more than 1.5 and 3 points (see 6.1 & 6.2). As mentioned in chapter 4, the content of the corpora differs in its discourse type and sentence structure. The legal-dcep and legal-jrc-acquis do contain enumerations, sentence fragments and cross references to a higher degree compared to the legal-europarl. These syntactical differences are directly reflected in the evaluation across the test sets derived from the very same corpora. It is becoming apparent that even in the single-task scenario, the multi-task model can cope better with the nature of the legal-dcep (descriptive legal text) and legal-jrc-acquis (legislative documents). The MultiModel Light falls off and does not reach the same level in the BLEU metric. Still, the training step count was sufficient for the MultiModel Light to converge. Therefore, the trade-off in model capacity over a reduced hidden and filter size come into play and cause distinct lower BLEU scores. In contrast to the BLEU metric, the MultiModel Light performs relatively better in terms of the CHRF metric which focuses more on recall than precision. Looking at the decodes and comparing translations manually, characteristics of the models become apparent. Occasionally, the MultiModel Light introduces small grammatical faults and vocabulary mistakes (see table 6.3) in comparison to the MultiModel Base and Transformer Base.

In addition, the MultiModel Light produces shorter sentences on average, which does not compulsory lead towards worse translations (see table 6.4). Manual inspection and concrete examples show that the translations are actually good across all models despite miscellaneous appearing BLEU scores on the metrics. The shorter sentences of the MultiModel Light and sentence structure variations of the Transformer Base are fined, whereas the semantic remains largely untouched. Besides, few mistakes are made by all models, e.g. character case faults. Reference translations are sometimes partially independent from their input (see first sentence part of the input and reference in table 6.3) which impact the scores but not the quality of the translations. Obviously, the models adhere to the input in a greater extent than a human translator which is fully aware of the context. In what sense this characteristic aids in legal environments is up to be determined. Anyway, present legal discourse types and complex sentences do not pose a specific challenge either to the MultiModel or the Transformer. The abundant training data for legal translation certainly contributes at this juncture.

Variances can be observed in the translation across languages. First, scores in translating from German to other languages differ considerably with English as target language at the top and Swedish at the bottom. A cause in the language roots of the

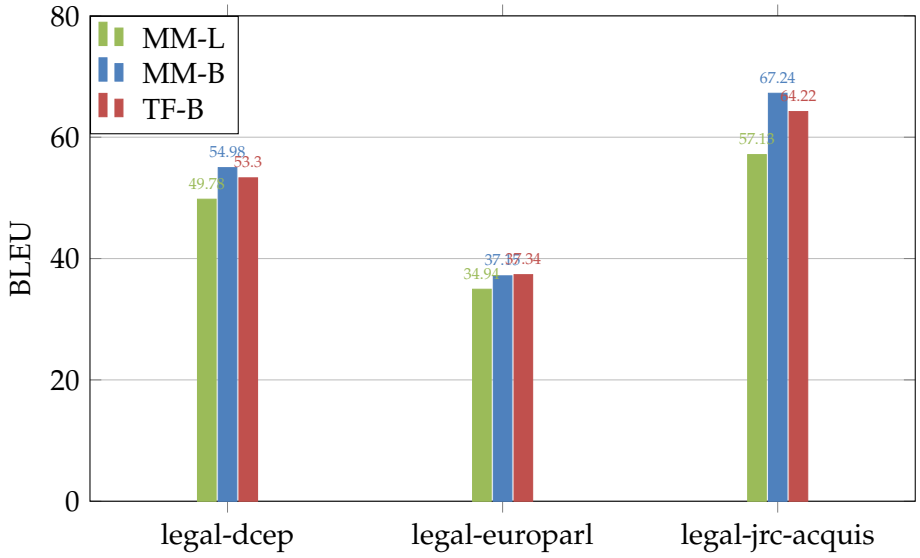


Figure 6.1.: German-to-English single-task translation performance of the MultiModel Light (MM-L), MultiModel Base (MM-B) and Transformer Base (TF-B) - BLEU

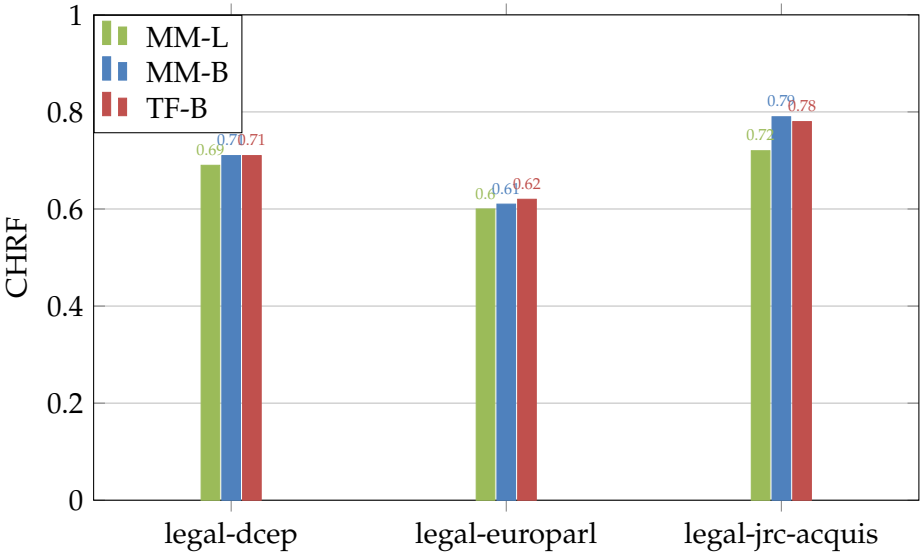


Figure 6.2.: German-to-English single-task translation performance of the MultiModel Light (MM-L), MultiModel Base (MM-B) and Transformer Base (TF-B) - CHRF

	BLEU	Example
Input	-	Im Gegensatz zu den Prophezeiungen von verschiedener Seite hat sich diese Position nach den Anschlägen vom 11. September nicht gewandelt.
MM-L	43.93	Unlike the prophesion of various sides, this position has not changed after the attacks of 11 September.
MM-B	44.48	Contrary to the prophecies of various quarters, this position has not changed since the attacks of 11 September.
TF-B	30.60	This position has not changed following the attacks of 11 September, contrary to the statements made by various quarters.
Reference	-	Contrary to what some people predicted, this position has not altered following the attacks of 11 September.

Table 6.3.: Single-task translation examples of the legal-europarl by the MultiModel Light (MM-L), MultiModel Base (MM-B) and Transformer Base (TF-B)

	BLEU	Example
Input	-	9 . Argentinien gewährleistet die Einhaltung dieser Vereinbarung insbesondere dadurch , daß es innerhalb der in dieser Vereinbarung festgelegten Mengen Ausfuhrlizenzen für die unter Nummer 1 genannten Erzeugnisse erteilt .
MM-L	17.61	9. Argentina shall ensure compliance with this Agreement by granting the export licences referred to in point 1 within the quantities laid down in this Agreement.
MM-B	29.63	9. Argentina shall ensure compliance with this Agreement, in particular by issuing export licences for the products referred to in point 1 within the quantities specified in this Agreement.
TF-B	40.09	9. Argentina shall ensure compliance with this Agreement in particular by issuing export licences for the products referred to in point 1 within the limits of the quantities laid down in this Agreement.
Reference	-	9. Argentina shall ensure that this arrangement is observed, in particular, by issuing export certificates covering the products referred to in paragraph 1 within the limits of the quantities covered by this arrangement.

Table 6.4.: Single-task translation examples of the legal-jrc-acquis by the MultiModel Light (MM-L), MultiModel Base (MM-B) and Transformer Base (TF-B)

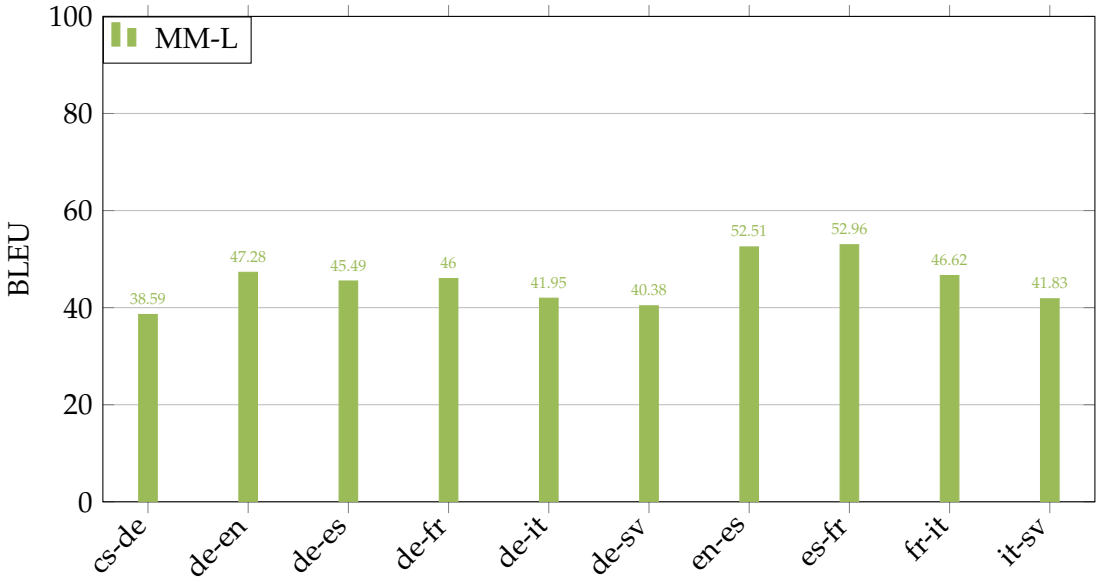


Figure 6.3.: Mean scores across corpora (legal-dcep, legal-europarl, legal-jrc-acquis) of the MultiModel Light (MM-L) - BLEU

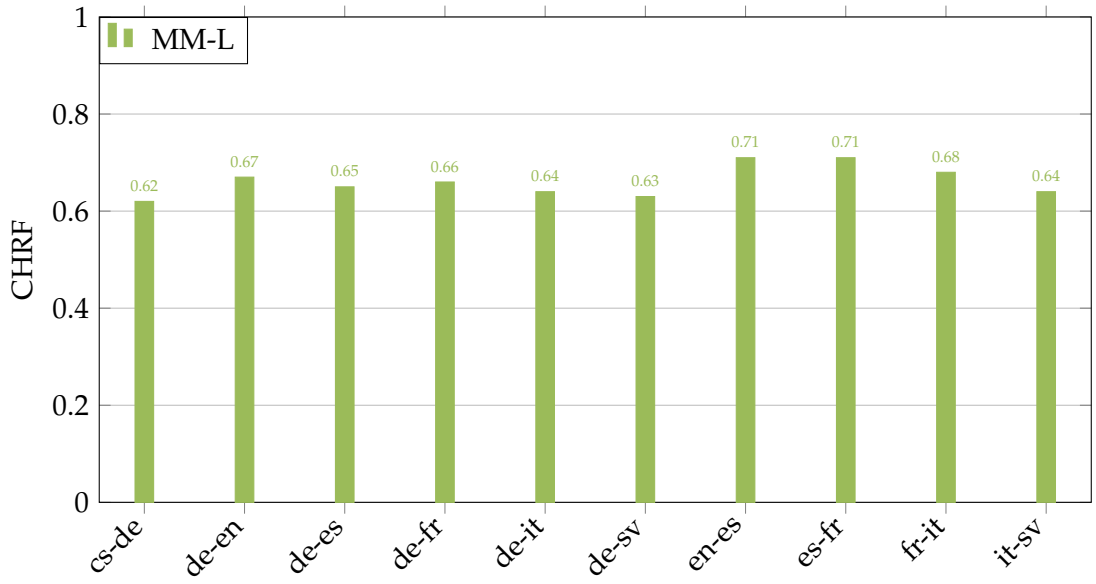


Figure 6.4.: Mean scores across corpora (legal-dcep, legal-europarl, legal-jrc-acquis) of the MultiModel Light (MM-L) - CHRF

		legal-dcep			legal-europarl			legal-jrc-acquis		
		MM-L	MM-B	TF-B	MM-L	MM-B	TF-B	MM-L	MM-B	TF-B
cs-de	BLEU	45.06	-	-	25.70	-	-	45.01	-	-
	CHRF	0.66	-	-	0.55	-	-	0.66	-	-
de-en	BLEU	49.78	**54.98**	53.30	34.94	37.15	**37.34**	57.13	**67.24**	64.22
	CHRF	0.69	**0.71**	**0.71**	0.60	0.61	**0.62**	0.72	**0.79**	0.78
de-es	BLEU	48.69	-	-	32.06	-	-	55.72	-	-
	CHRF	0.67	-	-	0.57	-	-	0.71	-	-
de-fr	BLEU	47.63	-	-	33.90	-	-	56.48	-	-
	CHRF	0.67	-	-	0.58	-	-	0.72	-	-
de-it	BLEU	44.37	-	-	27.08	-	-	54.40	-	-
	CHRF	0.66	-	-	0.55	-	-	0.71	-	-
de-sv	BLEU	43.65	-	-	26.08	-	-	51.42	-	-
	CHRF	0.65	-	-	0.55	-	-	0.68	-	-
en-es	BLEU	53.66	-	-	42.65	-	-	61.21	-	-
	CHRF	0.72	-	-	0.66	-	-	0.76	-	-
es-fr	BLEU	53.20	-	-	39.84	-	-	65.84	-	-
	CHRF	0.71	-	-	0.63	-	-	0.80	-	-
fr-it	BLEU	48.53	-	-	32.17	-	-	59.16	-	-
	CHRF	0.70	-	-	0.59	-	-	0.76	-	-
it-sv	BLEU	43.24	-	-	26.32	-	-	55.94	-	-
	CHRF	0.65	-	-	0.56	-	-	0.72	-	-

Table 6.5.: Single-task translation performance of the MultiModel Light (MM-L), MultiModel Base (MM-B) and Transformer Base (TF-B)

languages (Germanic, Romanic, Slavik) cannot be distinguished. Rather, the language pair and target language tend to influence the legal translation. Second, all tasks with German as source language score visibly worse compared to tasks translating to same target languages with another source language (see figure 6.3 & 6.4). It can therefore be concluded, that German is less suitable as source language than other languages at learning translation in a multi-language setting.

6.2.2. Summarization

The MultiModel Base was not included in single-task summarization experiments. Therefore, a comparison is made between the MultiModel Light and Transformer Base. The Transformer Base performs perceivably worse than the MultiModel Light in summarization (see figure 6.5). Surely, summarization is a text-to-text task, however it is becoming apparent that despite being a related task to translation, it demands varying capabilities from a model. Hence, the Transformer architecture may not be adequate for summarization. Though, it cannot be ruled out that an increase in the Transformer model's performance is probably achievable through tuning the hyperparameters, still improvements will be within limits. These summarization results clearly favor the MultiModel and its objective to provide a generic architecture to cover a variety of use cases. The MultiModel does not only fit multi-task requirements, its light version also outperforms the Transformer Base in legal summarization. Differences across languages can also be observed (see 6.6). Though, they are not as large as in translation. We also observed differences in reference summary length across languages, with Czech being the language with the shortest summaries against German and French with the longest summaries.

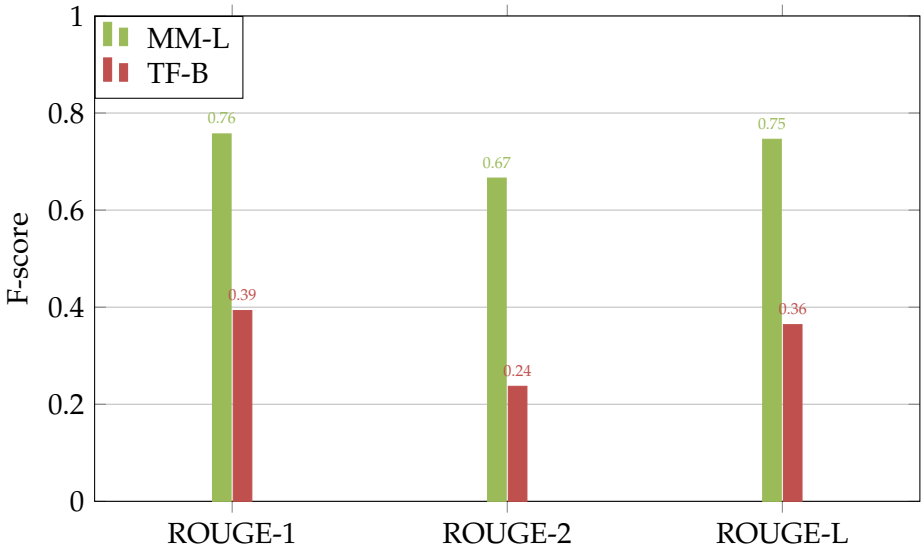


Figure 6.5.: German single-task summarization performance of the MultiModel Light (MM-L) and Transformer Base (TF-B) - F-score

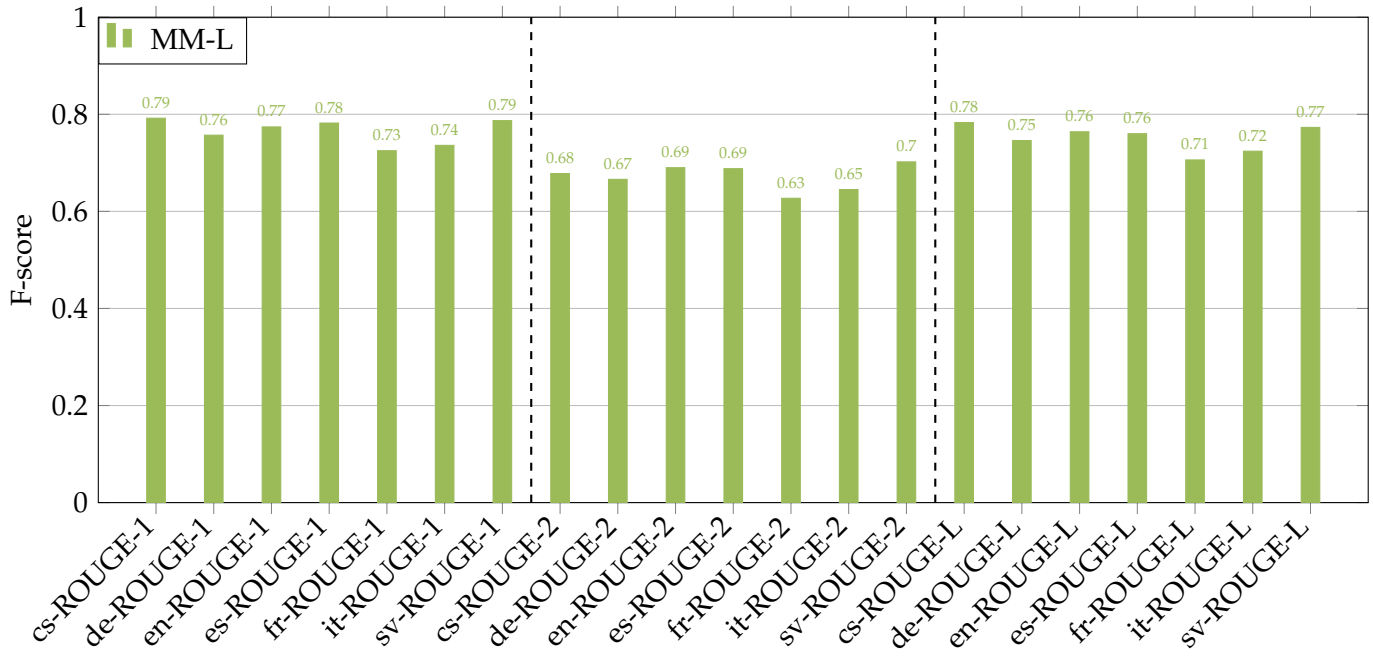


Figure 6.6.: Single-task summarization performance of the MultiModel Light (MM-L) across languages - F-score

		ROUGE-1		ROUGE-2		ROUGE-L	
		MM-L	TF-B	MM-L	TF-B	MM-L	TF-B
cs	Recall	0.792	-	0.678	-	0.796	-
	Precision	0.805	-	0.689	-	0.782	-
	F-score	0.792	-	0.678	-	0.783	-
de	Recall	**0.743**	0.400	**0.654**	0.240	**0.732**	0.370
	Precision	**0.798**	0.422	**0.700**	0.255	**0.783**	0.390
	F-score	**0.757**	0.393	**0.666**	0.237	**0.746**	0.364
en	Recall	0.758	-	0.677	-	0.749	-
	Precision	0.833	-	0.743	-	0.822	-
	F-score	0.774	-	0.690	-	0.764	-
es	Recall	0.772	-	0.678	-	0.750	-
	Precision	0.816	-	0.716	-	0.793	-
	F-score	0.782	-	0.688	-	0.760	-
fr	Recall	0.716	-	0.618	-	0.698	-
	Precision	0.769	-	0.662	-	0.748	-
	F-score	0.725	-	0.627	-	0.706	-
it	Recall	0.722	-	0.633	-	0.709	-
	Precision	0.773	-	0.677	-	0.756	-
	F-score	0.736	-	0.645	-	0.724	-
sv	Recall	0.782	-	0.698	-	0.769	-
	Precision	0.820	-	0.727	-	0.804	-
	F-score	0.787	-	0.702	-	0.773	-

Table 6.6.: Single-task summarization performance of the MultiModel Light (MM-L) and Transformer Base (TF-B)

6.2.3. Multi-label Classification

We trained the EuroVoc document classification task analogously to the summarization experiments for 100k steps. Again, we left out the MultiModel Base from single-task multi-label classification due to lacking time. During the experiments with the Transformer Base, the measures have shown that it is not capable to learn this classification task to an acceptable state with a maximum of 5% accuracy. In order to be able to apply sequence-to-sequence models to this task, we changed the output conditions upfront 5.2.1. Hence, the circumstances already pointed towards this result. In contrast, the MultiModel Light is capable to learn this type of classification task despite the output adjustment.

We compare the single-task results of the MultiModel Light with the JRC EuroVoc Indexer JEX [6] (see figure 6.7). While achieving a higher precision on average, the MultiModel Light lacks in recall and yields lower F1 scores for all languages except English and French (see 6.7). Mentionable, the JRC EuroVoc Indexer JEX was cross-validated over the whole dataset while models within this work were evaluated on a consistent test set. Concluding, we have set a baseline with the single task results which we use to evaluate the performance of various multi-task combinations.

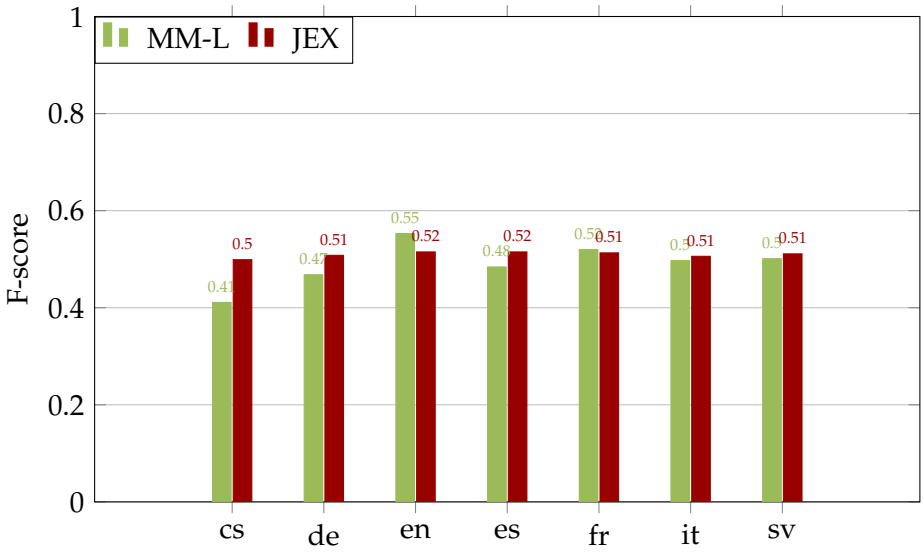


Figure 6.7.: Single-task multi-label classification performance of the MultiModel Light (MM-L) across languages - F-score

		MM-L single	JRC EuroVoc Indexer JEX
cs	Accuracy	0.366	-
	Recall	0.408	**0.521**
	Precision	0.413	**0.469**
	F-score	0.411	**0.493**
	Atleast 1	0.708	-
de	Accuracy	0.422	-
	Recall	0.465	**0.473**
	Precision	0.471	**0.549**
	F-score	0.468	**0.519**
	Atleast 1	0.759	-
en	Accuracy	0.493	-
	Recall	0.543	**0.555**
	Precision	**0.563**	0.480
	F-score	**0.553**	0.523
	Atleast 1	0.854	-
es	Accuracy	0.437	-
	Recall	0.476	**0.555**
	Precision	**0.493**	0.480
	F-score	0.484	**0.519**
	Atleast 1	0.774	-
fr	Accuracy	**0.463**	-
	Recall	0.509	**0.554**
	Precision	**0.532**	0.478
	F-score	**0.520**	0.513
	Atleast 1	**0.845**	-
it	Accuracy	**0.441**	-
	Recall	0.485	**0.546**
	Precision	0.509	0.471
	F-score	0.497	**0.506**
	Atleast 1	0.812	-
sv	Accuracy	**0.438**	-
	Recall	0.483	**0.547**
	Precision	**0.521**	0.479
	F-score	0.501	**0.511**
	Atleast 1	**0.792**	-

Table 6.7.: Single-task multi-label classification performance of the MultiModel Light (MM-L) and JRC EuroVoc Indexer JEX [6]

6.3. Multi-Task Training

The comparison between multi-task learning and single-task learning on the MultiModel are the main focus of our experiments. Instead of a detailed comparison with the Transformer model, we propose additional combinations by joining together different tasks to find hidden synergies. In particular, we wanted to show whether multi-task deep learning can be beneficial for tasks in the legal domain. In the following sections, we show how training simultaneously on multiple legal tasks compare to training on each task separately and deduce answers to posed research questions.

6.3.1. Translation

For legal translation, the possible combinations are numerous since preprocessed language pairs and their reversals constitute to the selection pool. We picked out two combinations, which seemed likely to reveal synergies. First, we propose a combination containing translation tasks with the same source language in order to improve generalization by jointly translating into different target languages. We call this combination jt-pool-5. We selected all five available German translation pairs and trained the MultiModel Light and MultiModel Base jointly on this combination. Analogously to the single task training, the training step count was 500k.

The MultiModel Light trained on the jt-pool-5 combination performed about 10% worse in the BLEU metric throughout all German translation tasks compared to training the tasks separately (see figure 6.8). No deviation in the relative difference can be observed between corpora and involved languages. It becomes clear that the jt-pool-5 combination does not put forth expected benefits in legal translation performance on the light version of MultiModel. The reason for this result is likely found in the limited capacity of the MultiModel Light. The more tasks are joined together in a combination, the more capacity is generally required from the model. Naturally, each translation task demands a part of the network's capacity for exclusive representations which are not used by other tasks. By adding more tasks these exclusive areas grow in numbers, besides having shared representations which also likely grow in size. If the model's capacity is too small to accommodate necessary task-specific and shared representations, it will not learn tasks to their full extent. To investigate the impact of the amount of translation tasks, we subsequently added tasks and evaluated the translation performance for each of the five resulting scenarios. By training concurrently, the phenomenon of deficient capacity can be observed over spikes in the evaluation during the training. The multi-task training may overwrite and distort already learned representations for a translation task by learning another one. The joint scores even temporarily exceed the single task scores, but are corrected in the long run. Finally, this leads to the overall performance being successively lower with an increasing number of tasks (see figure 6.9).

The second applied combination comprises tasks joint in a chain-like manner to

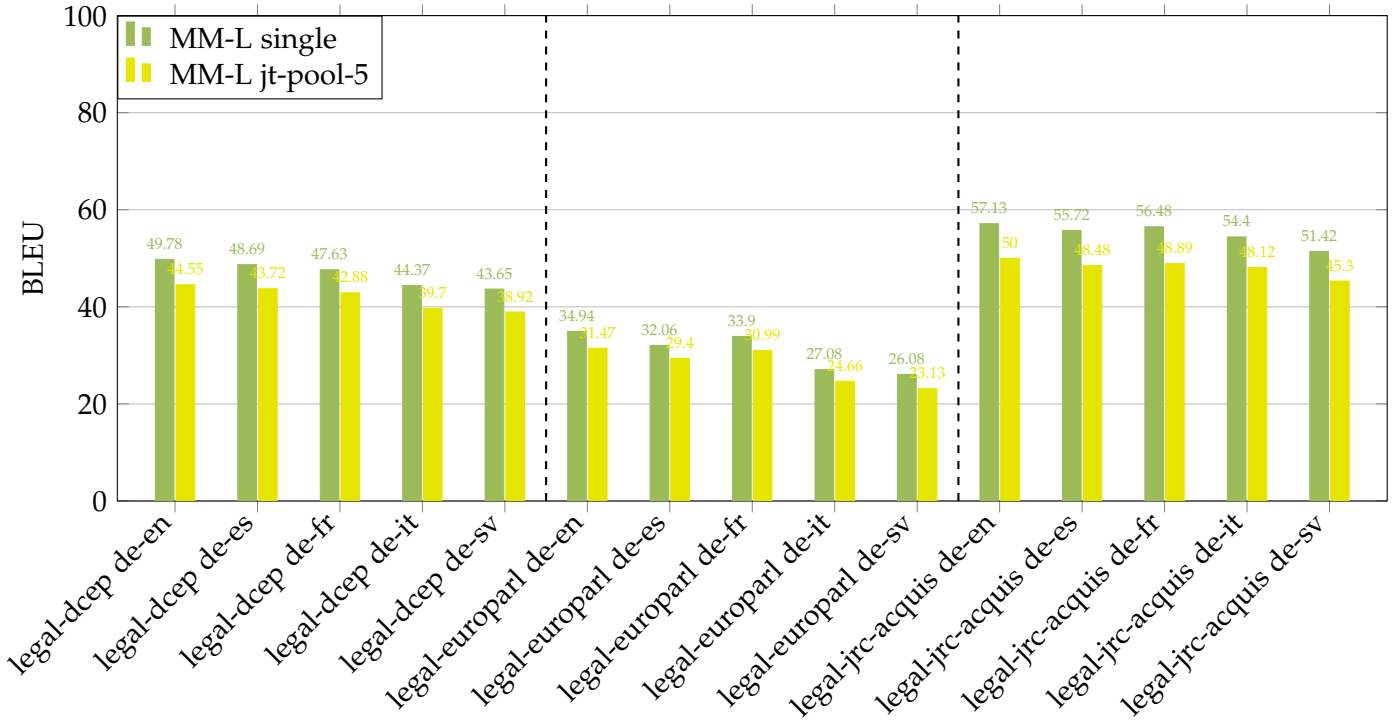


Figure 6.8.: Single-task & multi-task (jt-pool-5) translation performance of the Multi-Model Light (MM-L) - BLEU

possibly improve performance by alternation. Each language appears as source and target language once, except the language in the beginning and the language at the end of the chain. We call this combination jt-chain-7. Similar to the jt-pool combinations, the results suggest an insufficient capacity. Subsequently, the performance is on average 15% lower when training 7 tasks jointly compared to training them separately (see. figure 6.10). Except anticipated differences across corpora and language pairs, salient features between the combinations cannot be observed.

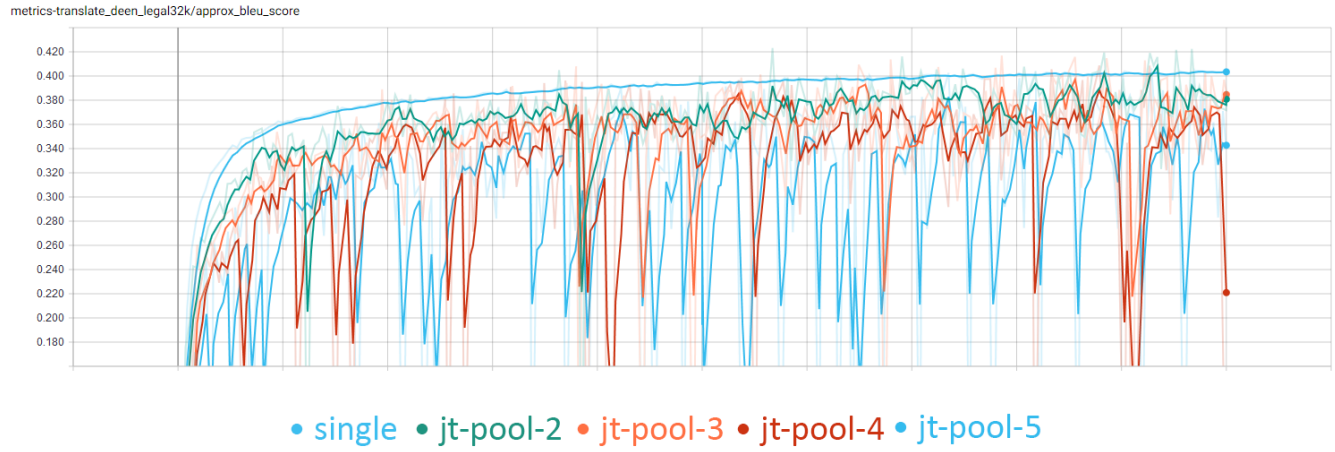


Figure 6.9.: Translation performance depending on the amount of tasks of the MultiModel Light (MM-L) - BLEU

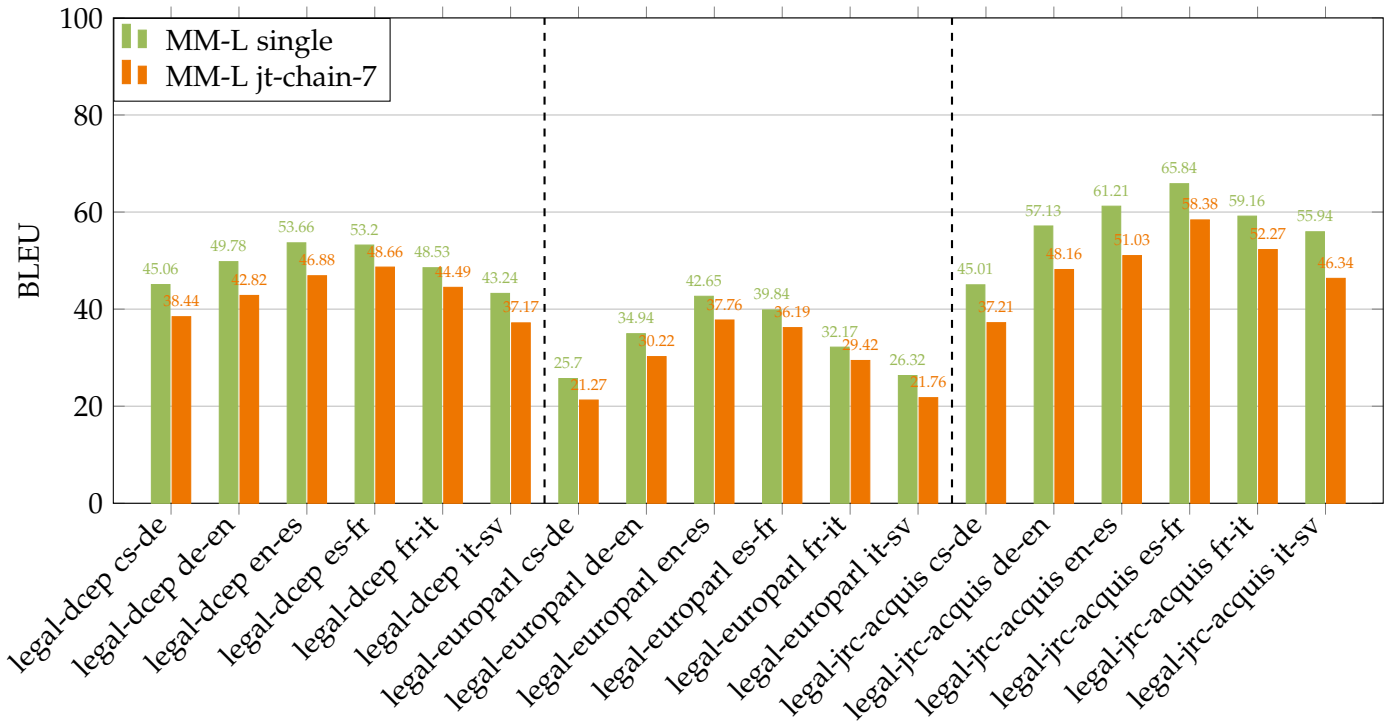


Figure 6.10.: Single-task & multi-task (jt-chain-7) translation performance of the Multi-Model Light (MM-L) - BLEU

As a consequence, we trained both translation combinations (jt-pool-5 & jt-chain-7) on the MultiModel Base. The MultiModel Base provides about 10 times more capacity than the MultiModel Light. This is directly reflected in the performance. Both joint combinations trained on the MultiModel Base outperform their single-task and multi-task counterparts on the MultiModel Light (see 6.12 & 6.13). This outcome is not surprising. The sheer capacity increase boosts the translation capabilities of the model. Howsoever, the results did not exceed their single-task training counterparts on the MultiModel Base (see figure 6.11 for the German-to-English BLEU scores). In addition, the Transformer Base does also perform better on all corpora, especially on the legal-europarl where it achieves the highest German-to-English BLEU score. The assumption arises that the capacity of the MultiModel is still not enough. Unfortunately, we could not further enlarge the model and leave this investigation to future research. We can conclude from these multi-task translation experiments on legal corpora, that capacity is essential for performance when joining a number of legal translation tasks. An answer to research question 4 definitely includes the advice to opt for larger models at the current point in time. We also show that the tested combinations consisting of multiple translation tasks do not yield improvements for legal translation with the selected models and hyperparameter sets compared to single-task training. Nonetheless, we find hitherto benefits of multi-task legal translation in shorter training times (see table 6.1).

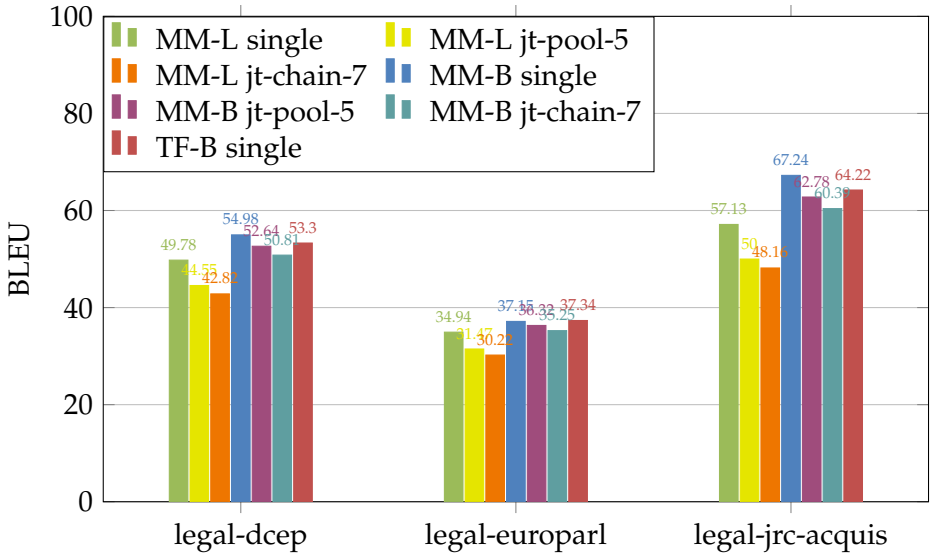


Figure 6.11.: German-to-English translation performance of single-task & multi-task translation combinations trained on the MultiModel Light (MM-L), MultiModel Base (MM-B) and Transformer Base (TF-B) - BLEU

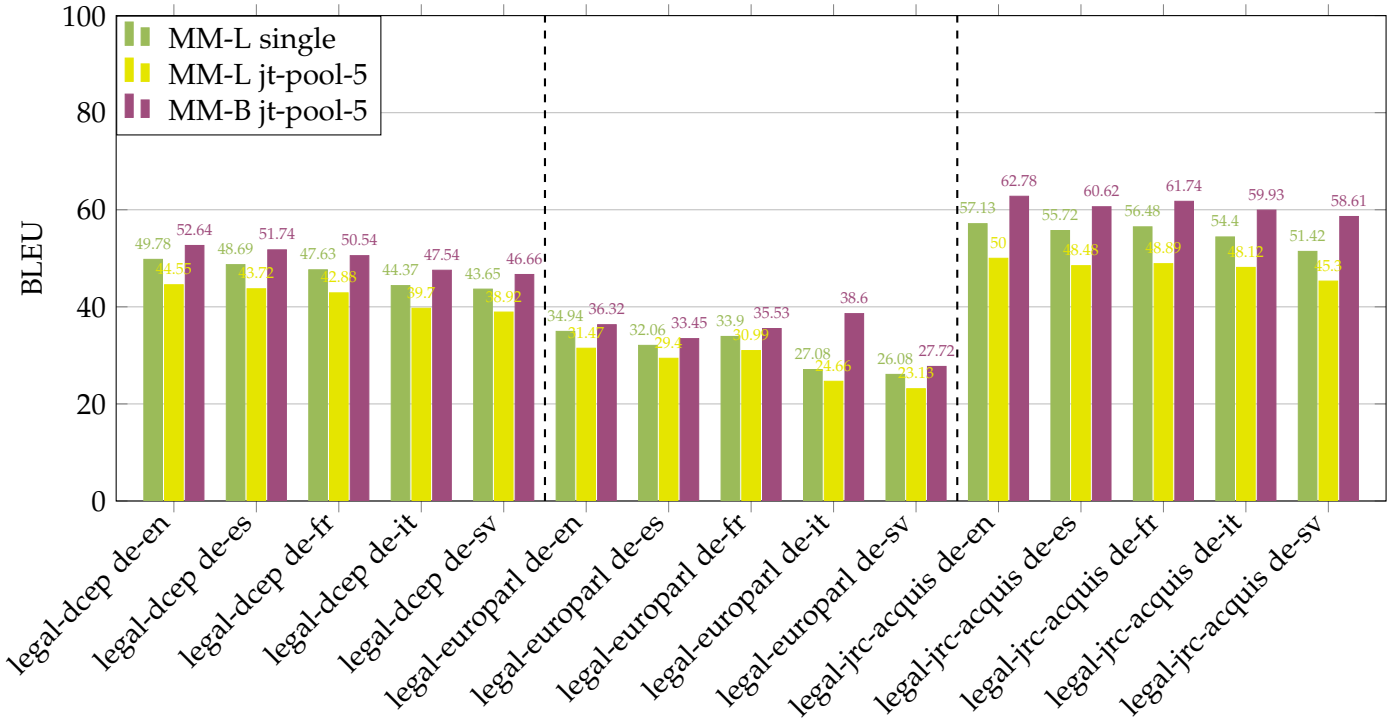


Figure 6.12.: Single-task & multi-task (jt-pool-5) translation performance of the Multi-Model Light - BLEU

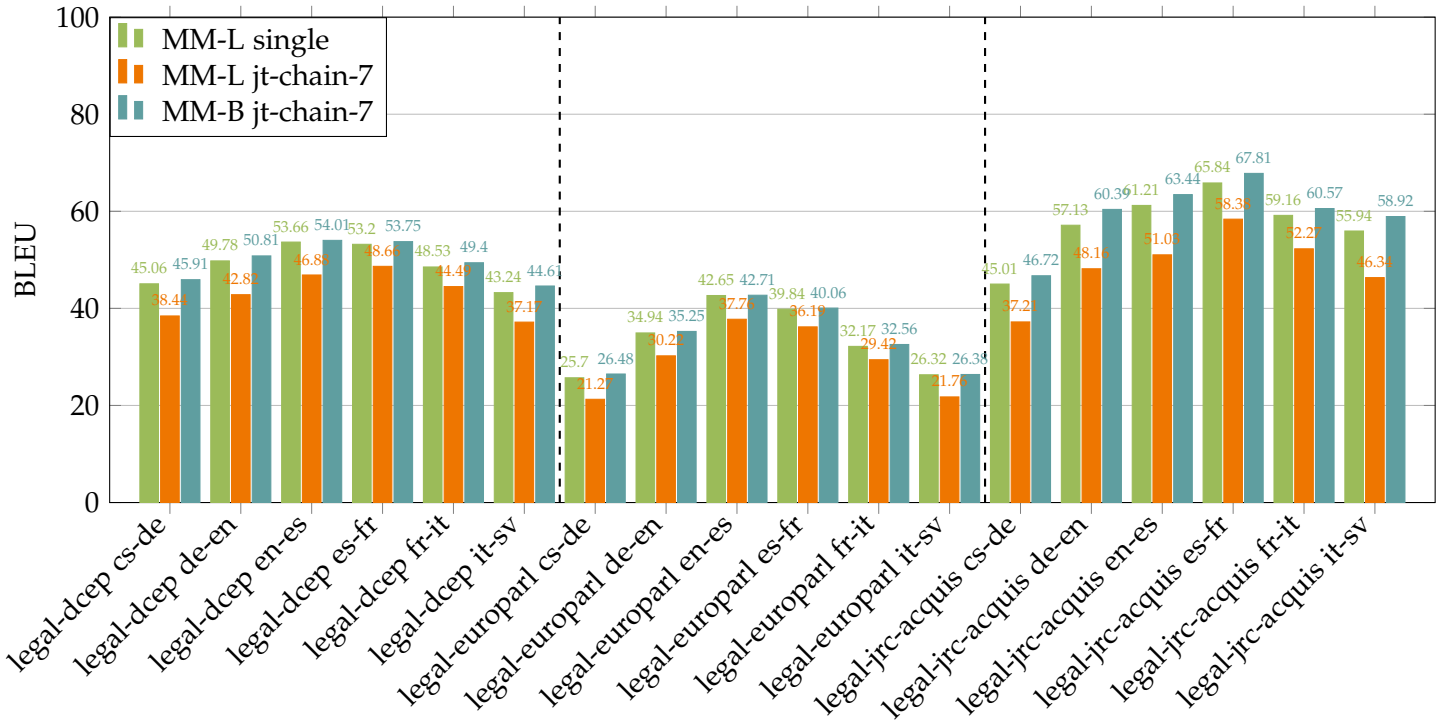


Figure 6.13.: Multi-task (jt-chain-7) & single-task translation performance of the Multi-Model Light (MM-L) - BLEU

6.3.2. Summarization

The multi-task legal summarization experiments also focused on the MultiModel Light. We included a single combination in our experiments, the joint training of all 7 tasks (js-7). We compare the MultiModel Light with single and joint training to the Transformer Base in the German summarization task. The single-task MultiModel Light scores the highest on all ROUGE metrics, whereas the MultiModel Light js-7 scores between the single variant and the Transformer Base. The separately trained model shows to be capable to extract more information from the full texts (see examples in 6.8). When looking closely at the ROUGE results, the gap for summarizing Czech documents is not as large compared to all other languages between joint and single training (see figure 6.15). This can be explained though much shorter reference summaries across the Czech part of the corpora which often do only contain a concise document title. Once more, separate training outperforms its joint training counterpart. Again, we infer the limit in the small capacity of the MultiModel Light.

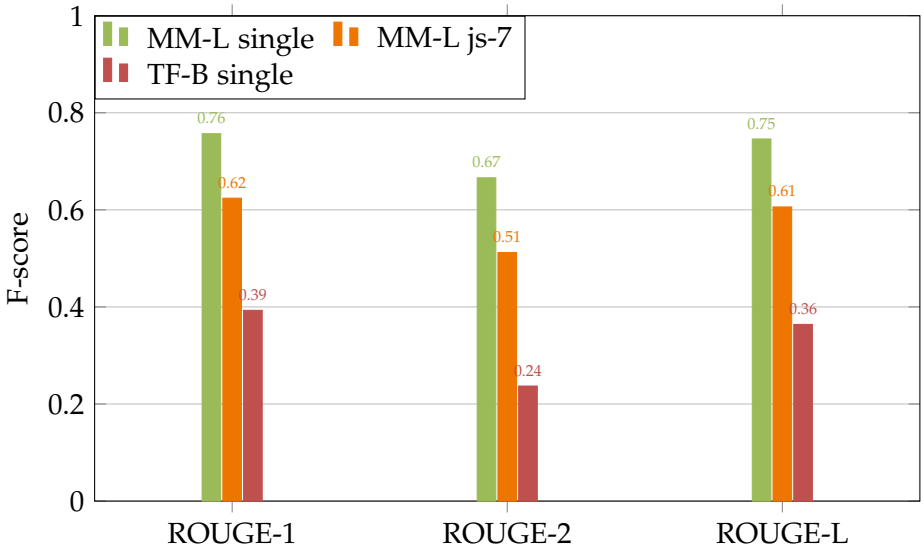


Figure 6.14.: Single-task & multi-task (js-7) summarization performance of the MultiModel Light (MM-L) and Transformer-Base (TF-B) - BLEU

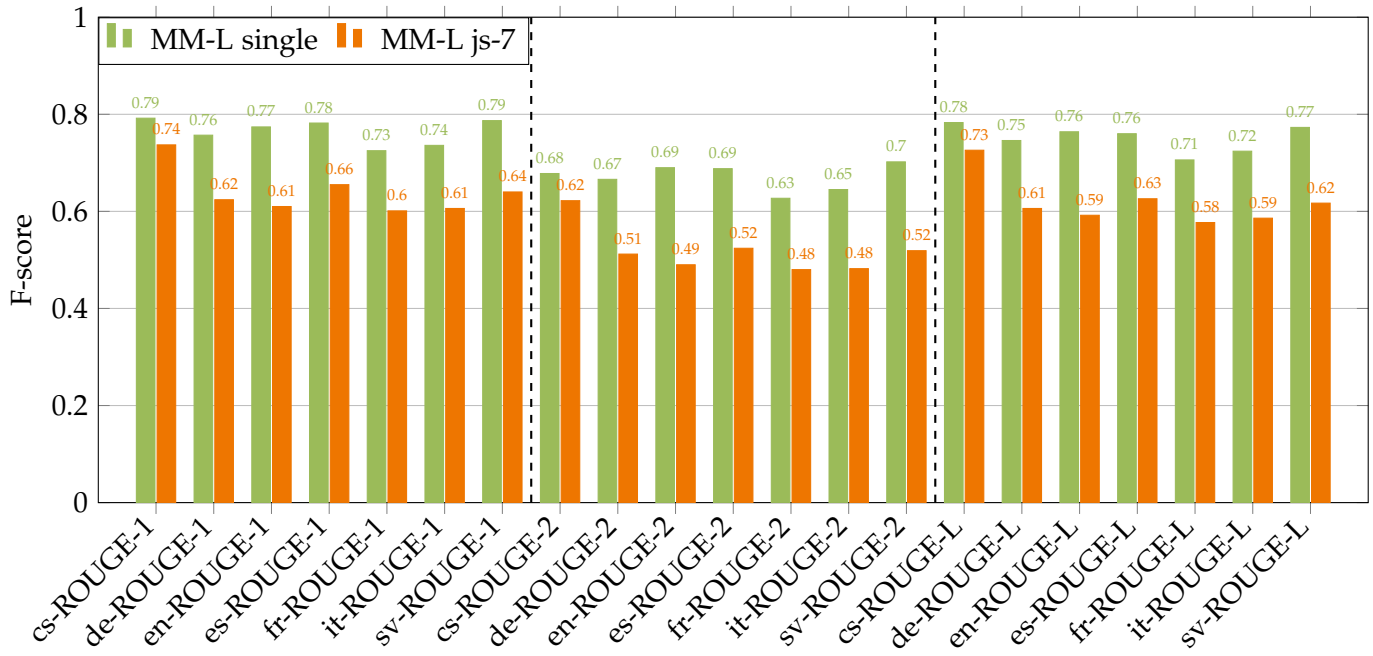


Figure 6.15.: Single-task & multi-task (js-7) summarization performance of the Multi-Model Light (MM-L) - F-score

2. Product The product under review is aluminium foil of a thickness of not less than 0,009 mm and not more than 0,018 mm, not backed, not further worked than rolled, in reels of a width not exceeding 650 mm originating in Russia (the product concerned), normally declared within CN code ex76071110. This CN code is given only for information. 3. Existing measures The measures currently in force are definitive anti-dumping duties imposed by Council Regulation (EC) No 950/2001 [2] on imports of aluminium foil originating, inter alia, in Russia. 4. Grounds for the review The request pursuant to Article 11(3) is based on the prima facie evidence that the circumstances on the basis of which measures were established have changed and that these changes are of a lasting nature. The applicant alleges and provides evidence showing that a comparison of normal value based on its own costs and prices, and export prices to the EU, would lead to a reduction of dumping significantly below the level of the current measures. Therefore, the continued imposition of measures at the existing levels, which were based on the level of dumping previously established, is no longer necessary to offset dumping. 5. Procedure for the determination of dumping Having determined, after consulting the Advisory Committee, that sufficient evidence exists to justify the initiation of a partial interim review, the Commission hereby initiates a review in accordance with Article 11(3) of the basic Regulation limited in scope to the examination of dumping as far as the applicant is concerned. The investigation will assess the need for the continuation, removal or amendment of the existing measures in respect of the applicant. (a) Questionnaires In order to obtain the information it deems necessary for its investigation, the Commission will send questionnaires to the applicant and to the authorities of the exporting country concerned. This information and supporting evidence should reach the Commission within the time limit set in point 6(a) of this notice. (b) Collection of information and holding of hearings All interested parties are hereby invited to make their views known, submit information other than questionnaire replies and to provide supporting evidence. This information and supporting evidence must reach the Commission within the time limit set in paragraph 6(a) of this notice. Furthermore, the Commission may hear interested parties, provided that they make a request showing that there are particular reasons why they should be heard. This request must be made within the time limit set in paragraph 6(b) of this notice. 6. Time limits (a) For parties to make themselves known, to submit questionnaire replies and any other information All interested parties, if their representations are to be taken into account during the investigation, must make themselves known by contacting the Commission, present their views and submit questionnaire replies or any other information within 40 days of the date of publication of this notice in the Official Journal of the European Union, unless otherwise specified. Attention is drawn to the fact that the exercise of most procedural rights set out in the basic Regulation depends on the party's making itself known within the aforementioned period. (b) Hearings All interested parties may also apply to be heard by the Commission within the same 40-day time limit. 7. Written submissions, questionnaire replies and correspondence All submissions and requests made by interested parties must be made in writing (not in electronic format, unless otherwise specified) and must indicate the name, address, e-mail address, telephone and fax, and/or telex numbers of the interested party. All written submissions, including the information requested in this notice, questionnaire replies and correspondence provided by interested parties on a confidential basis shall be labelled as "Limited" [3] and, in accordance with Article 19(2) of the basic Regulation, shall be accompanied by a non-confidential version, which will be labelled "For inspection by interested parties". Commission address for correspondence: European Commission Directorate General for Trade Directorate B Office: J-79 5/16 B-1049 Brussels Fax (32-2) 295 65 05 Telex COMEU B 21877 8. Non-cooperation In cases in which any interested party refuses access to or otherwise does not provide the necessary information within the time limits, or significantly impedes the investigation, findings, affirmative or negative, may be made in accordance with Article 18 of the basic Regulation, on the basis of the facts available. Where it is found that any interested party has supplied false or misleading information, the information shall be disregarded and use may be made, in accordance with Article 18 of the basic Regulation, of the facts available. If an interested party does not cooperate, or cooperates only partially, and use of the facts available is made, the result may be less favourable to that party than if it had cooperated. ——— [1] OJ L 56, 6.3.1996, p. 1. Regulation as last amended by Council Regulation (EC) No 461/2004 (OJ L 77, 13.3.2004, p.12). [2] OJ L 134,17.5.2001, p. 1. Regulation as last amended by Council Regulation (EC) No 998/2004 (OJ L 183, 20.5.2004, p.4). See also Notice 2004/C 193/03 (OJ C 193, 29.7.2004, p.3) concerning the modification of the name and address of Open Joint Stock Company Rusal Sayanal. [3] This means that the document is for internal use only. It is protected pursuant to Article 4 of Regulation (EC) No 1049/2001 of the European Parliament and of the Council (OJ L 145, 31.5.2001, p. 43). It is a confidential document pursuant to Article 19 of Council Regulation (EC) No 384/96 (OJ L 56, 6.3.1996, p.1) and Article 6 of the Agreement on Implementation of Article VI of the GATT 1994 (Anti-dumping Agreement).

MM-L single	Notice of initiation of a partial interim review of the antidumping measures applicable to imports of television camera systems originating in Russia
MM-L js-7	Notice concerning the anti-dumping measures
Reference	Notice of initiation of a partial interim review of the anti-dumping measures applicable to imports of certain aluminium foil originating in Russia

Table 6.8.: Single-task & multi-task (js-7) summarization example om the MultiModel Light (MM-L) from the legal-jrc-acquis-summarize

6.3.3. Multi-label Classification

The EuroVoc classification task yields the most diverse results. In previous single-task experiments, we showed the MultiModel Light is capable to perform on the same level than the JRC EuroVoc Indexer JEX [6], while generally reaching higher precision adverse to recall. Subsequently, we joined together all 7 multi-label classification tasks (jl-7) and trained the MultiModel Light in order to see the difference in model performance specifically to this special task. The results show that the joint combination is indeed capable to outperform the single-task variant in the Czech language across all metrics when trained on the same amount of training steps (see figure 6.16). Additionally, the precision of the joint task is higher compared to the single task in more than half of all tasks 6.18. We infer the increased performance in the Czech classification by the joint task through the size of the training set. The Czech task falls back to 1.5k documents less than other languages (see 4.12). Through joint training, the model is capable of transfer learning by having access to other tasks and indirectly to more training data. On the other side, this happens not to be the only the reason. The Swedish task has the least documents available. Nonetheless, the single task outperforms the joint task on all metrics. Indications lead to varying requirements in classifying documents of a specific language. The MultiModel Light joint task obviously copes better with the Czech language, while it falls behind specifically in the Swedish language. In this regard, the joint task also reaches the highest precision in the English, Spanish and Italian language. Analogously to the single-task comparison with JEX, the joint task reaches higher precision across all classification tasks but lacks recall against JEX. All numbers can be seen at the end of this section in table 6.18.

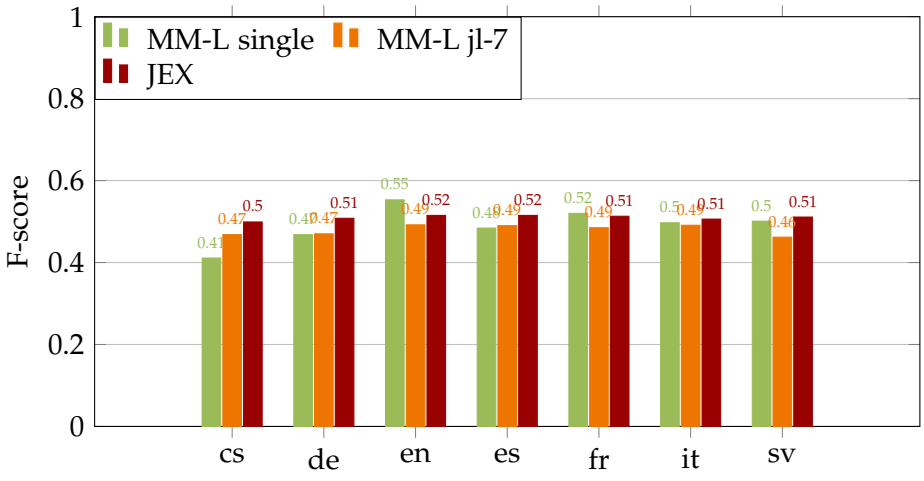


Figure 6.16.: Single-task & multi-task (jl-7) multi-label classification performance of the MultiModel Light (MM-L) and JRC EuroVoc Indexer JEX [6] - F-score

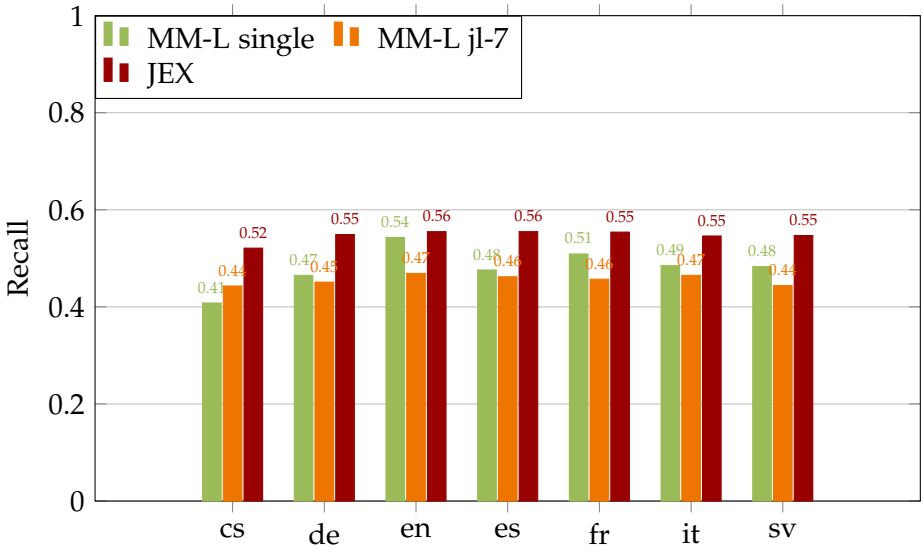


Figure 6.17.: Single-task & multi-task (jl-7) multi-label classification performance of the MultiModel Light (MM-L) and JRC EuroVoc Indexer JEX [6] - Recall

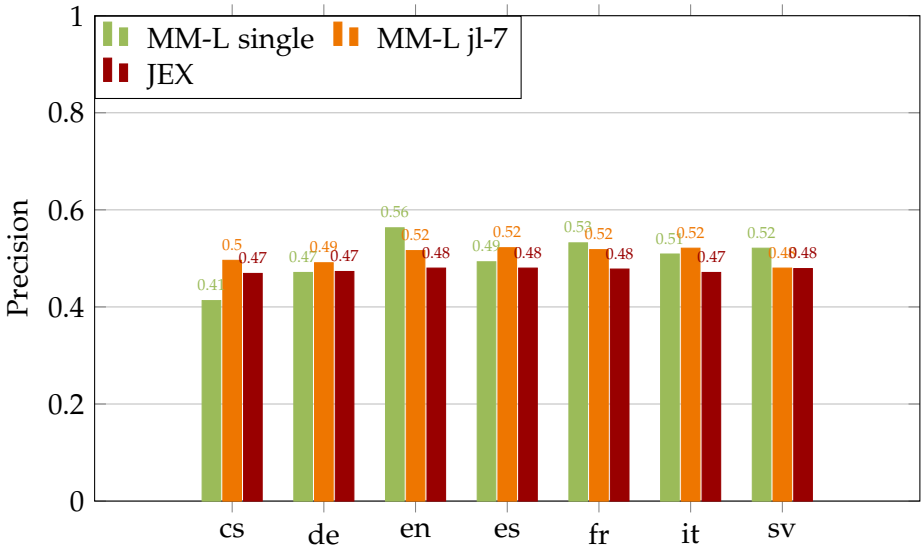


Figure 6.18.: Single-task & multi-task (jl-7) multi-label classification performance of the MultiModel Light (MM-L) and JRC EuroVoc Indexer JEX [6] - Precision

6.3.4. Across Task Families

The final included joint training in our experiments is a combination across task families. We join together the German-to-English translation, German summarization and German multi-label classification task (ja-3). We train this joint task for 500k steps. The intention is to further improve the single-task performance by training tasks jointly with seemingly independent tasks. Similar to the experiments conducted with the publication of the MultiModel, an increased performance through such unusual combinations may be possible. We compare this joint task with the single-task and previous multi-task results.

In the German-to-English translation task, the ja-3 trained on the MultiModel Light performs worse than the single-task equivalents (see figure 6.19). However, it scores better than the jt-pool-5 and jt-chain-7 combination. Interestingly, the ja-3 beats the jt-pool-3 combination on the legal-dcep corpus (see table 6.11). Therefore, joining across unrelated tasks can be more beneficial than joining the same amount of related tasks in legal translation. In the German legal summarization task, the ja-3 trained on the MultiModel Light performs better than the joint task across all seven summarization tasks (js-7), but worse than the single task counterpart (see figure 6.20). This additionally shows that the amount of tasks in a joint task has a much higher impact to performance than the relatedness or diversity of the joined tasks. Regarding the German multi-label classification task, the ja-3 combination trained on the MultiModel Light performs exceptionally well. It outperforms the single-task variant, the full joint task over 7 multi-label classification tasks (jl-7) and JEX on all metrics except recall (see figure 6.21 & table 6.18). Again, we show that multi-task deep learning can be beneficial in the German legal document classification. The training set of the classification task is by far not as large as the training set for the German-to-English translation task. The addition of the translation task as well as the summarization task and indirectly their training data facilitates transfer learning across task families even on the smaller version of the MultiModel. Anyway, insufficient capacity of the MultiModel Light is still the issue for mixed results amongst all tasks.

Finally, we trained the ja-3 combination on the MultiModel Base to verify the results of the light version. The ja-3 joint task is the combination with the least amount of tasks joined together trained on the MultiModel Base. Therefore, the capacity issue is mitigated and may fade into the background. The ja-3 joint task on the MultiModel Base yields the highest result in the German-to-English translation on the legal-dcep corpus across all single and multi-task training runs (see figure 6.19). Though, the gap to the MultiModel Base single task training is minimal and almost neglectable (about 0.13 BLEU). However, we show that by joining fewer tasks and combining tasks across task families the performance of the MultiModel Base reaches equal heights adverse to single-task training. Beyond that, the ja-3 trained on the MultiModel Base yields overall highest results in German summarization and German multi-label classification (see figure 6.20 & 6.21). It exceeds the JRC EuroVoc Indexer JEX by 14 points on the F1

metric. The capacity issue appears to be resolved through joining less tasks. Further, we show that more efficient transfer learning takes place through joining across task families. We conclude our experiments with tables recapitulating all collected results (6.11, 6.12, 6.13, 6.14, 6.15, 6.16, 6.17, 6.18).

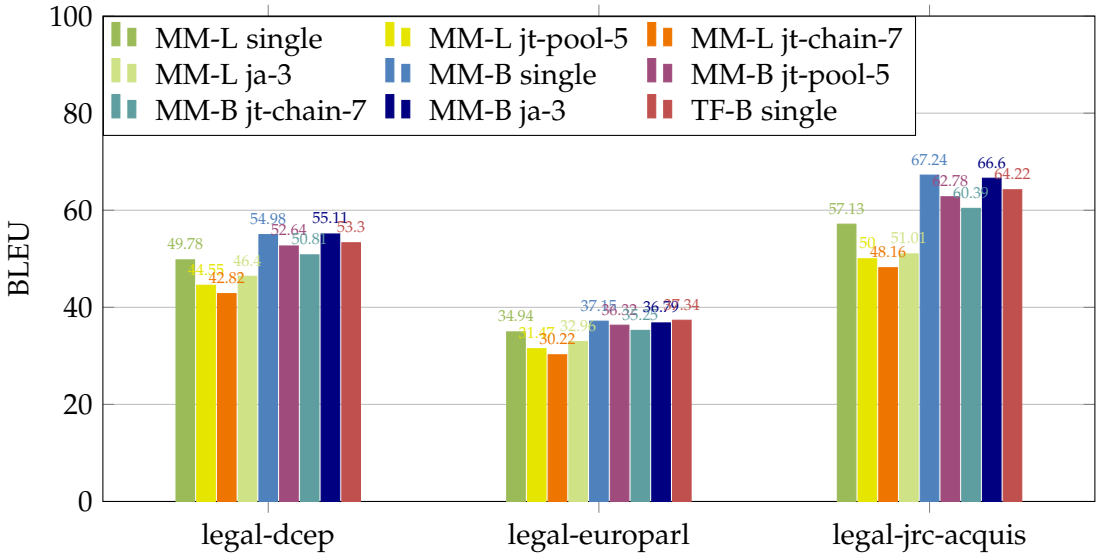


Figure 6.19.: Final German-to-English translation performance of all single-task & multi-task translation combinations trained on the MultiModel Light (MM-L), MultiModel Base (MM-B) and Transformer Base (TF-B) - BLEU

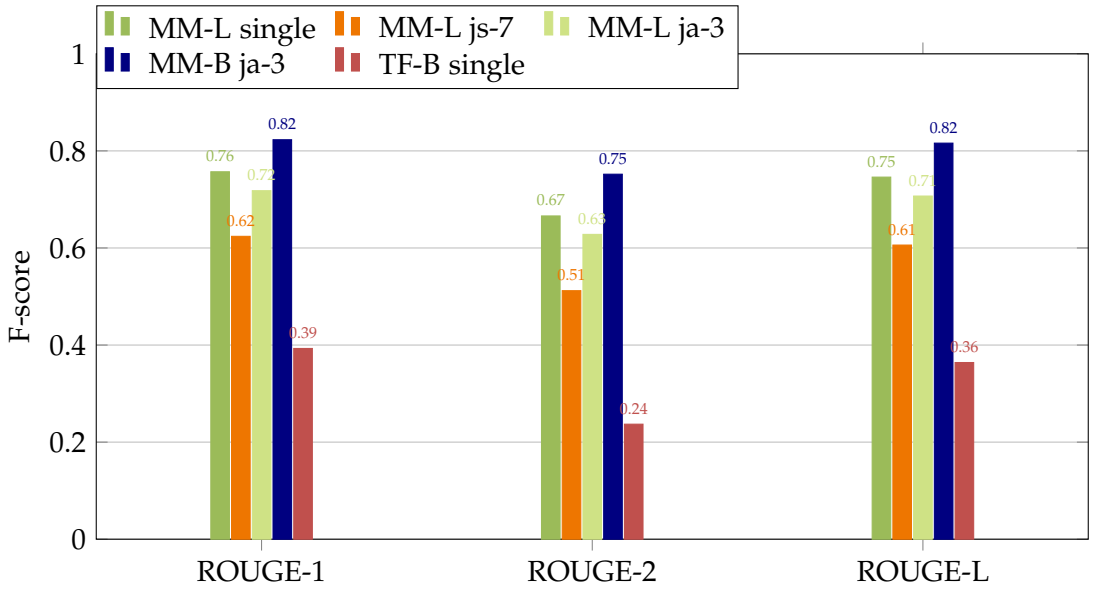


Figure 6.20.: Single-task & multi-task (js-7, ja-3) summarization performance of the MultiModel Light (MM-L), MultiModel Base (MM-B) and Transformer Base (TF-B) - F-score

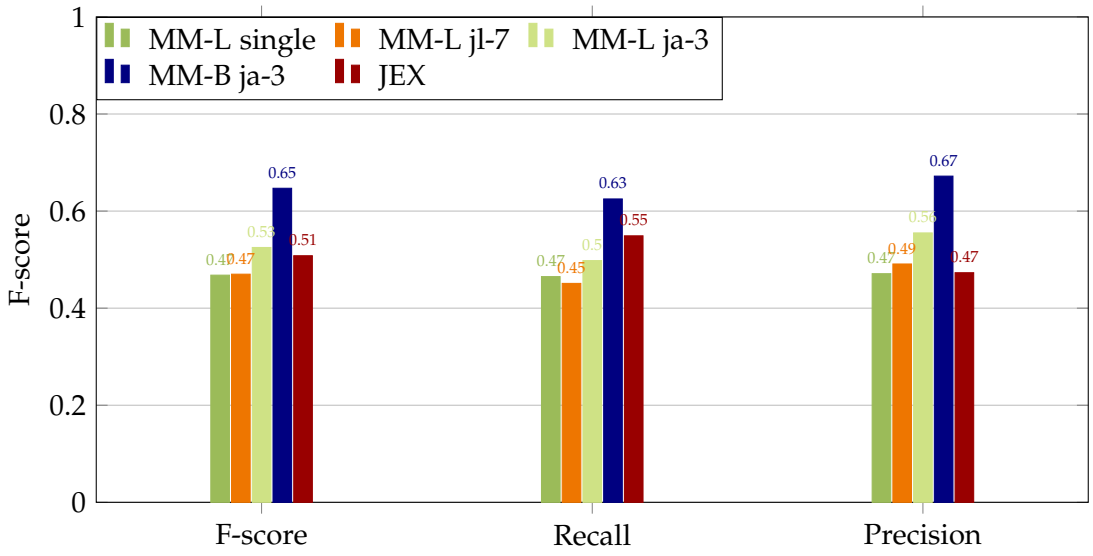


Figure 6.21.: German single-task & multi-task (jl-7, ja-3) multi-label classification performance of the MultiModel Light (MM-L), MultiModel Base (MM-B) and JRC EuroVoc Indexer JEX [6] - F-score, Recall, Precision

	BLEU	Example
Input	-	9 . Argentinien gewährleistet die Einhaltung dieser Vereinbarung insbesondere dadurch , daß es innerhalb der in dieser Vereinbarung festgelegten Mengen Ausfuhrlizenzen für die unter Nummer 1 genannten Erzeugnisse erteilt .
MM-L single	17.61	9. Argentina shall ensure compliance with this Agreement by granting the export licences referred to in point 1 within the quantities laid down in this Agreement.
MM-L jt-pool-5	27.57	9. Argentina shall ensure compliance with this Agreement, in particular by providing for export licences for the products referred to in point 1 within the quantities set out in this Agreement.
MM-L jt-chain-7	34.01	9. Argentina shall ensure compliance with this Agreement, in particular by granting export licences for products referred to in paragraph 1 within the quantities set out in this Agreement.
MM-L ja-3	25.72	9. Argentina shall ensure compliance with this Agreement in particular by granting export licences for the products referred to in point 1 within the quantities laid down in this Agreement.
MM-B single	29.63	9. Argentina shall ensure compliance with this Agreement, in particular by issuing export licences for the products referred to in point 1 within the quantities specified in this Agreement.
MM-B jt-pool-5	30.02	9. Argentina shall ensure compliance with this Agreement, in particular by granting it export licences for the products referred to in point 1 within the limits laid down in this Agreement.
MM-B jt-chain-7	29.71	9. Argentina shall ensure compliance with this Agreement, in particular by issuing export licences for the products referred to in point 1 within the quantities set out in this Agreement.
MM-B ja-3	50.62	9. Argentina shall ensure compliance with this Agreement, in particular by issuing export licences for the products referred to in paragraph 1 within the limits of the quantities fixed in this Agreement.
TF-B single	40.09	9. Argentina shall ensure compliance with this Agreement in particular by issuing export licences for the products referred to in point 1 within the limits of the quantities laid down in this Agreement.
Reference	-	9. Argentina shall ensure that this arrangement is observed, in particular, by issuing export certificates covering the products referred to in paragraph 1 within the limits of the quantities covered by this arrangement.

Table 6.9.: German-to-English translation examples of the legal-jrc-acquis for all trained combinations

Identifier	Tasks
jt-pool-2	German-to-English translation, German-to-French translation
jt-pool-3	German-to-English translation, German-to-French translation, German-to-Italian translation
jt-pool-4	German-to-English translation, German-to-French translation, German-to-Italian translation, German-to-Spanish translation
jt-pool-5	German-to-English translation, German-to-French translation, German-to-Italian translation, German-to-Spanish translation, German-to-Swedish translation
jt-chain-7	Czech-to-German translation, German-to-English translation, English-to-Spanish translation, Spanish-to-French translation, French-to-Italian translation, Italian-to-Swedish translation
js-7	Czech summarization, German summarization, English summarization, Spanish summarization, French summarization, Italian summarization, Swedish summarization
jl-7	Czech multi-label classification, German multi-label classification, English multi-label classification, Spanish multi-label classification, French multi-label classification, Italian multi-label classification, Swedish multi-label classification
ja-3	German-to-English translation, German summarization, German multi-label classification

Table 6.10.: Multi-task combinations of legal-dcep, legal-europarl, legal-jrc-acquis, legal-jrc-acquis-summarize, legal-jrc-acquis-label

legal-dcep								
		MM-L single	MM-L jt-pool-5	MM-L jt-pool-4	MM-L jt-pool-3	MM-L jt-pool-2	MM-L jt-chain-7	MM-L ja-3
cs-de	BLEU	**45.06**	-	-	-	-	38.44	-
	CHRF	**0.66**	-	-	-	-	0.61	-
de-en	BLEU	**49.78**	44.55	44.96	45.99	47.17	42.82	46.40
	CHRF	**0.69**	0.65	0.65	0.66	0.67	0.64	0.66
de-es	BLEU	**48.69**	43.72	44.03	45.09	45.96	-	-
	CHRF	**0.67**	0.64	0.64	0.65	0.66	-	-
de-fr	BLEU	**47.63**	42.88	43.35	44.34	-	-	-
	CHRF	**0.67**	0.64	0.64	0.65	-	-	-
de-it	BLEU	**44.37**	39.70	40.23	-	-	-	-
	CHRF	**0.66**	0.63	0.63	-	-	-	-
de-sv	BLEU	**43.65**	38.92	-	-	-	-	-
	CHRF	**0.65**	0.62	-	-	-	-	-
en-es	BLEU	**53.66**	-	-	-	-	46.88	-
	CHRF	**0.72**	-	-	-	-	0.68	-
es-fr	BLEU	**53.20**	-	-	-	-	48.66	-
	CHRF	**0.71**	-	-	-	-	0.68	-
fr-it	BLEU	**48.53**	-	-	-	-	44.49	-
	CHRF	**0.70**	-	-	-	-	0.67	-
it-sv	BLEU	**43.24**	-	-	-	-	37.17	-
	CHRF	**0.65**	-	-	-	-	0.61	-

Table 6.11.: Single-task & multi-task translation performance of the MultiModel Light (MM-L) on the legal-dcep

legal-europarl								
		MM-L single	MM-L jt-pool-5	MM-L jt-pool-4	MM-L jt-pool-3	MM-L jt-pool-2	MM-L jt-chain-7	MM-L ja-3
cs-de	BLEU	**25.70**	-	-	-	-	21.27	-
	CHRF	**0.55**	-	-	-	-	0.52	-
de-en	BLEU	**34.94**	31.47	31.87	32.66	33.52	30.22	32.96
	CHRF	**0.60**	0.57	0.57	0.58	0.59	0.56	0.58
de-es	BLEU	**32.06**	29.40	29.74	30.27	30.91	-	-
	CHRF	**0.57**	0.55	0.56	0.56	**0.57**	-	-
de-fr	BLEU	**33.90**	30.99	31.43	32.06	-	-	-
	CHRF	**0.58**	0.56	0.56	0.57	-	-	-
de-it	BLEU	**27.08**	24.66	24.95	-	-	-	-
	CHRF	**0.55**	0.53	0.53	-	-	-	-
de-sv	BLEU	**26.08**	23.13	-	-	-	-	-
	CHRF	**0.55**	0.53	-	-	-	-	-
en-es	BLEU	**42.65**	-	-	-	-	37.76	-
	CHRF	**0.66**	-	-	-	-	0.62	-
es-fr	BLEU	**39.84**	-	-	-	-	36.19	-
	CHRF	**0.63**	-	-	-	-	0.60	-
fr-it	BLEU	**32.17**	-	-	-	-	29.42	-
	CHRF	**0.59**	-	-	-	-	0.57	-
it-sv	BLEU	**26.32**	-	-	-	-	21.76	-
	CHRF	**0.56**	-	-	-	-	0.52	-

Table 6.12.: Single-task & multi-task translation performance of the MultiModel Light (MM-L) on the legal-europarl

legal-jrc-acquis								
		MM-L single	MM-L jt-pool-5	MM-L jt-pool-4	MM-L jt-pool-3	MM-L jt-pool-2	MM-L jt-chain-7	MM-L ja-3
cs-de	BLEU	**45.01**	-	-	-	-	37.21	-
	CHRF	**0.66**	-	-	-	-	0.59	-
de-en	BLEU	**57.13**	50.00	50.89	52.15	54.11	48.16	51.01
	CHRF	**0.71**	0.67	0.68	0.69	0.70	0.66	0.67
de-es	BLEU	**55.72**	48.48	49.75	51.13	52.90	-	-
	CHRF	**0.71**	0.66	0.67	0.68	0.69	-	-
de-fr	BLEU	**56.48**	48.89	49.78	51.59	-	-	-
	CHRF	**0.72**	0.66	0.67	0.68	-	-	-
de-it	BLEU	**54.40**	48.12	48.97	-	-	-	-
	CHRF	**0.71**	0.66	0.67	-	-	-	-
de-sv	BLEU	**51.42**	45.30	-	-	-	-	-
	CHRF	**0.68**	0.64	-	-	-	-	-
en-es	BLEU	**61.21**	-	-	-	-	51.03	-
	CHRF	**0.76**	-	-	-	-	0.70	-
es-fr	BLEU	**65.84**	-	-	-	-	58.38	-
	CHRF	**0.80**	-	-	-	-	0.75	-
fr-it	BLEU	**59.16**	-	-	-	-	52.27	-
	CHRF	**0.76**	-	-	-	-	0.72	-
it-sv	BLEU	**55.94**	-	-	-	-	46.34	-
	CHRF	**0.72**	-	-	-	-	0.66	-

Table 6.13.: Single-task & multi-task translation performance of the MultiModel Light (MM-L) on the legal-jrc-acquis

legal-dcep									
		MM-B single	TF-B single	MM-B jt-pool-5	MM-B jt-pool-4	MM-B jt-pool-3	MM-B jt-pool-2	MM-B jt-chain-7	MM-B ja-3
cs-de	BLEU	x	x	-	-	-	-	45.91	-
	CHRF	x	x	-	-	-	-	0.67	-
de-en	BLEU	54.98	53.30	52.64	x	x	x	50.81	**55.11**
	CHRF	0.71	0.71	0.71	x	x	x	0.69	**0.72**
de-es	BLEU	x	x	51.74	x	x	x	-	-
	CHRF	x	x	0.70	x	x	x	-	-
de-fr	BLEU	x	x	50.54	x	x	-	-	-
	CHRF	x	x	0.69	x	x	-	-	-
de-it	BLEU	x	x	47.54	x	-	-	-	-
	CHRF	x	x	0.68	x	-	-	-	-
de-sv	BLEU	x	x	46.66	-	-	-	-	-
	CHRF	x	x	0.67	-	-	-	-	-
en-es	BLEU	x	x	-	-	-	-	54.01	-
	CHRF	x	x	-	-	-	-	0.72	-
es-fr	BLEU	x	x	-	-	-	-	53.75	-
	CHRF	x	x	-	-	-	-	0.72	-
fr-it	BLEU	x	x	-	-	-	-	49.40	-
	CHRF	x	x	-	-	-	-	0.70	-
it-sv	BLEU	x	x	-	-	-	-	44.41	-
	CHRF	x	x	-	-	-	-	0.66	-

Table 6.14.: Single-task & multi-task translation performance of the MultiModel Base (MM-B) and Transformer Base (TF-B) on the legal-dcep

legal-europarl									
		MM-B single	TF-B single	MM-B jt-pool-5	MM-B jt-pool-4	MM-B jt-pool-3	MM-B jt-pool-2	MM-B jt-chain-7	MM-B ja-3
cs-de	BLEU	x	x	-	-	-	-	26.48	-
	CHRF	x	x	-	-	-	-	0.56	-
de-en	BLEU	37.15	**37.34**	36.32	x	x	x	35.25	36.79
	CHRF	0.61	**0.62**	0.61	x	x	x	0.60	0.61
de-es	BLEU	x	x	33.45	x	x	x	-	-
	CHRF	x	x	0.59	x	x	x	-	-
de-fr	BLEU	x	x	35.53	x	x	-	-	-
	CHRF	x	x	0.59	x	x	-	-	-
de-it	BLEU	x	x	28.60	x	-	-	-	-
	CHRF	x	x	0.56	x	-	-	-	-
de-sv	BLEU	x	x	27.72	-	-	-	-	-
	CHRF	x	x	0.56	-	-	-	-	-
en-es	BLEU	x	x	-	-	-	-	42.71	-
	CHRF	x	x	-	-	-	-	0.66	-
es-fr	BLEU	x	x	-	-	-	-	40.06	-
	CHRF	x	x	-	-	-	-	0.63	-
fr-it	BLEU	x	x	-	-	-	-	32.56	-
	CHRF	x	x	-	-	-	-	0.59	-
it-sv	BLEU	x	x	-	-	-	-	26.38	-
	CHRF	x	x	-	-	-	-	0.56	-

Table 6.15.: Single-task & multi-task translation performance of the MultiModel Base (MM-B) and Transformer Base (TF-B) on the legal-europarl

legal-jrc-acquis									
		MM-B single	TF-B single	MM-B jt-pool-5	MM-B jt-pool-4	MM-B jt-pool-3	MM-B jt-pool-2	MM-B jt-chain-7	MM-B ja-3
cs-de	BLEU	x	x	-	-	-	-	46.72	-
	CHRF	x	x	-	-	-	-	0.67	-
de-en	BLEU	**67.24**	64.22	62.78	x	x	x	60.39	66.60
	CHRF	**0.79**	0.78	0.77	x	x	x	0.75	**0.79**
de-es	BLEU	x	x	60.62	x	x	x	-	-
	CHRF	x	x	0.75	x	x	x	-	-
de-fr	BLEU	x	x	61.74	x	x	-	-	-
	CHRF	x	x	0.76	x	x	-	-	-
de-it	BLEU	x	x	59.93	x	-	-	-	-
	CHRF	x	x	0.75	x	-	-	-	-
de-sv	BLEU	x	x	58.61	-	-	-	-	-
	CHRF	x	x	0.74	-	-	-	-	-
en-es	BLEU	x	x	-	-	-	-	63.44	-
	CHRF	x	x	-	-	-	-	0.78	-
es-fr	BLEU	x	x	-	-	-	-	67.81	-
	CHRF	x	x	-	-	-	-	0.81	-
fr-it	BLEU	x	x	-	-	-	-	60.57	-
	CHRF	x	x	-	-	-	-	0.77	-
it-sv	BLEU	x	x	-	-	-	-	58.92	-
	CHRF	x	x	-	-	-	-	0.75	-

Table 6.16.: Single-task & multi-task translation performance of the MultiModel Base (MM-B) and Transformer Base (TF-B) on the legal-jrc-acquis

legal-jrc-acquis-summarize																							
		ROUGE-1							ROUGE-2							ROUGE-L							
		MM-L single	MM-L js-7	MM-L ja-3	MM-B single	MM-B js-7	MM-B ja-3	TF-B single	MM-L single	MM-L js-7	MM-L ja-3	MM-B single	MM-B js-7	MM-B ja-3	TF-B single	MM-L single	MM-L js-7	MM-L ja-3	MM-B single	MM-B js-7	MM-B ja-3	TF-B single
cs	Recall	**0.792**	0.729	x	x	x	x	x	**0.678**	0.615	x	x	x	x	x	**0.796**	0.719	x	x	x	x	x
	Precision	**0.805**	0.786	x	x	x	x	x	**0.689**	0.664	x	x	x	x	x	**0.782**	0.773	x	x	x	x	x
	F-score	**0.792**	0.737	x	x	x	x	x	**0.678**	0.622	x	x	x	x	x	**0.783**	0.726	x	x	x	x	x
de	Recall	0.743	0.615	0.700	x	x	**0.809**	0.400	0.654	0.504	0.613	x	x	**0.739**	0.240	0.732	0.597	0.690	x	x	**0.802**	0.370
	Precision	0.798	0.673	0.780	x	x	**0.861**	0.422	0.700	0.551	0.672	x	x	**0.787**	0.255	0.783	0.653	0.768	x	x	**0.853**	0.390
	F-score	0.757	0.624	0.718	x	x	**0.823**	0.393	0.666	0.512	0.628	x	x	**0.752**	0.237	0.746	0.606	0.707	x	x	**0.816**	0.364
en	Recall	**0.758**	0.591	x	x	x	x	x	**0.677**	0.474	x	x	x	x	x	**0.749**	0.574	x	x	x	x	x
	Precision	**0.833**	0.701	x	x	x	x	x	**0.743**	0.555	x	x	x	x	x	**0.822**	0.681	x	x	x	x	x
	F-score	**0.774**	0.610	x	x	x	x	x	**0.690**	0.489	x	x	x	x	x	**0.764**	0.592	x	x	x	x	x
es	Recall	**0.772**	0.635	x	x	x	x	x	**0.678**	0.508	x	x	x	x	x	**0.750**	0.608	x	x	x	x	x
	Precision	**0.816**	0.719	x	x	x	x	x	**0.716**	0.575	x	x	x	x	x	**0.793**	0.687	x	x	x	x	x
	F-score	**0.782**	0.655	x	x	x	x	x	**0.688**	0.524	x	x	x	x	x	**0.760**	0.626	x	x	x	x	x
fr	Recall	**0.716**	0.584	x	x	x	x	x	**0.618**	0.467	x	x	x	x	x	**0.698**	0.561	x	x	x	x	x
	Precision	**0.769**	0.684	x	x	x	x	x	**0.662**	0.546	x	x	x	x	x	**0.748**	0.658	x	x	x	x	x
	F-score	**0.725**	0.601	x	x	x	x	x	**0.627**	0.480	x	x	x	x	x	**0.706**	0.577	x	x	x	x	x
it	Recall	**0.722**	0.603	x	x	x	x	x	**0.633**	0.478	x	x	x	x	x	**0.709**	0.582	x	x	x	x	x
	Precision	**0.773**	0.643	x	x	x	x	x	**0.677**	0.508	x	x	x	x	x	**0.756**	0.621	x	x	x	x	x
	F-score	**0.736**	0.607	x	x	x	x	x	**0.645**	0.482	x	x	x	x	x	**0.724**	0.586	x	x	x	x	x
sv	Recall	**0.782**	0.653	x	x	x	x	x	**0.698**	0.524	x	x	x	x	x	**0.769**	0.628	x	x	x	x	x
	Precision	**0.820**	0.663	x	x	x	x	x	**0.727**	0.537	x	x	x	x	x	**0.804**	0.638	x	x	x	x	x
	F-score	**0.787**	0.640	x	x	x	x	x	**0.702**	0.519	x	x	x	x	x	**0.773**	0.617	x	x	x	x	x

Table 6.17.: Single-task & multi-task summarization performance of the MultiModel Light (MM-L), MultiModel Light (MM-B) and Transformer Base (TF-B) on the legal-jrc-acquis-summarize

legal-jrc-acquis-label								
		MM-L single	MM-L jl-7	MM-L ja-3	MM-B single	MM-B jl-7	MM-B ja-3	JRC EuroVoc Indexer JEX
cs	Accuracy	0.366	**0.397**	x	x	x	x	-
	Recall	0.408	0.443	x	x	x	x	**0.521**
	Precision	0.413	**0.496**	x	x	x	x	0.469
	F-score	0.411	0.468	x	x	x	x	**0.499**
	Atleast 1	0.708	**0.791**	x	x	x	x	-
de	Accuracy	0.422	0.402	0.448	x	x	**0.586**	-
	Recall	0.465	0.451	0.498	x	x	**0.625**	0.549
	Precision	0.471	0.491	0.555	x	x	**0.672**	0.473
	F-score	0.468	0.470	0.525	x	x	**0.647**	0.508
	Atleast 1	0.759	0.807	0.837	x	x	**0.936**	-
en	Accuracy	**0.493**	0.421	x	x	x	x	-
	Recall	0.543	0.469	x	x	x	x	**0.555**
	Precision	**0.563**	0.516	x	x	x	x	0.480
	F-score	**0.553**	0.492	x	x	x	x	0.515
	Atleast 1	**0.854**	0.828	x	x	x	x	-
es	Accuracy	**0.437**	0.415	x	x	x	x	-
	Recall	0.476	0.462	x	x	x	x	**0.555**
	Precision	0.493	**0.522**	x	x	x	x	0.480
	F-score	0.484	0.490	x	x	x	x	**0.515**
	Atleast 1	0.774	**0.841**	x	x	x	x	-
fr	Accuracy	**0.463**	0.409	x	x	x	x	-
	Recall	0.509	0.457	x	x	x	x	**0.554**
	Precision	**0.532**	0.518	x	x	x	x	0.478
	F-score	**0.520**	0.485	x	x	x	x	0.513
	Atleast 1	**0.845**	0.822	x	x	x	x	-
it	Accuracy	**0.441**	0.413	x	x	x	x	-
	Recall	0.485	0.465	x	x	x	x	**0.546**
	Precision	0.509	**0.521**	x	x	x	x	0.471
	F-score	0.497	0.491	x	x	x	x	**0.506**
	Atleast 1	0.812	**0.836**	x	x	x	x	-
sv	Accuracy	**0.438**	0.398	x	x	x	x	-
	Recall	0.483	0.444	x	x	x	x	**0.547**
	Precision	**0.521**	0.480	x	x	x	x	0.479
	F-score	0.501	0.462	x	x	x	x	**0.511**
	Atleast 1	**0.792**	0.746	x	x	x	x	-

Table 6.18.: Single-task & multi-task multi-label classification performance of the MultiModel Light (MM-L), MultiModel Light (MM-B) and JRC EuroVoc Indexer JEX [6]

7. Conclusions

Concluding, we compiled several corpora for natural language processing tasks in the legal domain. This includes three ready-to-use legal translation corpora with a total of 158 million sentence pairs. In contrast to translation, the support for other legal tasks happens to be insufficient and is clearly visible in the amount of hitherto available legal corpora. Therefore, we produced a corpus for legal text summarization and legal document classification originating from the JRC-Acquis. Moreover, we presented a new corpus called Legal GCD consisting of German court decision documents issued by federal German courts. Including about 42k documents, the Legal GCD states a valuable data collection to encourage diverse tasks in the legal domain. With two additional derived corpora, we exhibit its application possibilities in document classification. Altogether, we took a first countermeasure against the data scarcity in the legal domain by simply providing more large annotated datasets suitable for data intense systems, especially for models based on deep artificial neural networks.

Building upon the compiled datasets, we integrated data generators into Tensor2Tensor to investigate the effects of multi-task deep learning in the legal domain. Through numerous experiments involving the MultiModel, we showed the impact of multi-task learning in the legal domain. For this purpose, we opposed the state-of-the-art multi-task model, the MultiModel, with two different hyperparameter sets to the base Transformer model. Multi-task combinations on the light version of the MultiModel performed poorer for translation and summarization in comparison to training tasks separately. We appoint the small capacity of the light version as the primary reason for these results. The only exception being the document classification task on the Legal JRC-Acquis Label corpus. Due to the relative small size of the corpus, multi-task learning across all languages achieved roughly coequal results despite joining together seven tasks. Multi-task learning is able to clearly outperform single-task learning in the Czech task which has a smaller amount of documents available compared to other languages. We prove that even the light variant of the MultiModel is capable of transfer learning, especially if single task data is tenuous which directly applies to many tasks in the legal domain. Further, we showed that the light version of the MultiModel progressively struggles when stepwise increasing the amount of joint tasks. We show the importance of high capacity in the multi-task environment and deduce that increasing capacity matters to multi-task learning in a greater extent obverse to single-task learning.

The comparison between the MultiModel Base and the Transformer Base yields

miscellaneous results. The Transformer Base and MultiModel Base perform close to each other in single-task legal translation. Against that, the Transformer Base performs poorly in summarization, while not even being capable to learn the multi-label classification to an acceptable state. Though, training these two tasks on the Transformer Base was experimental, since it is originally developed for translation. The multi-task results of the base version of the MultiModel also scored below single-task results, except for one combination which joined tasks across task families (German-to-English translation, German summarization and German multi-label classification). This combination just yields the highest BLEU score in the German-to-English translation task on the legal-dcep corpus amongst all trained models. In addition, the combination reaches an F1-score of 64.7 on the German multi-label classification task which is about 14 points higher than the score achieved by the JRC-Indexer JEX [6].

We showed that multi-task deep learning can be beneficial in the legal domain and conclude the following constraints. The amount of joined tasks plays a big role to the performance of the MultiModel and should be chosen accordingly to its capacity. Multi-task deep learning shows its effect clearly on tasks where data is sparse through indirectly bringing in more data from joined tasks. This way, it is possible for the model to outperform the single-task scenario and even beat state-of-the-art results. We found joint combinations across task families to have more potential than combinations within task families. This is especially interesting, since this outcome contradicts common intuitions. By far, we could not include many combinations in our experiments in order to get a broader view onto a lot of promising indications and leave them for further research.

8. Future Research

By providing ready-to-use corpora, a starting point is set for further work with data intense legal systems. Within this work, we could merely scratch the surface of multi-task deep learning and its applications in the legal domain. In the face of all tasks at our disposal, we could not test a major part of the multi-task combinations. E.g. a combination across language tasks with the same target language or a combination spanning over all translation tasks. The checkered results give definitely incentive to try out more combinations and investigate their performance. Moreover, the general addition of new legal corpora with various applicable tasks must be continued. But also, existing corpora have more to offer. The processing of additional language pairs (beyond 21) of the original DCEP and JRC-Acquis corpus was omitted which would open up even more possible task combinations.

Additionally, we conducted initial experiments with the Legal GCD (legal-gcd-court & legal-gcd-verdict). However, our tries ended up meaningless due to 5.2.1. A different approach to these classification tasks will likely yield better results.

On the other side, hyperparameters of the models were not specifically tuned. As follows, a wide range of different setups were not exhausted. Besides, innovative deep learning models are proposed frequently from which the legal domain can greatly profit. The application of these models needs to be driven further for resolving legal problems. Finally, joining across task families seems to be very promising. The authors of the MultiModel also reported an increase even by joining apparently unrelated tasks across domains. Hence, introducing datasets from other domains and combining them with legal datasets holds high potential to further improve multi-task deep learning and resolve the data scarcity in the legal domain. In conclusion, we set off the beginning of multi-task deep learning in the legal domain with our work and hope to instigate future research in this special area.

Appendices

A. Legal Corpora

Corpus	Legal	Translation	Summarization	Classification	Size
JRC-Acquis Corpus	✓	22 languages	✓	✓	463k documents
Digital Corpus of the European Parliament	✓	23 languages	-	-	1.5m documents
Europarl Corpus	✓	20 languages	-	-	30m sentences
Legal GCD	✓	german	-	✓	42k documents
DGT - Translation Memory	✓	24 languages	-	-	65m sentences
EAC - Translation Memory	✓	26 languages	-	-	78k sentences
MultiUN	✓	7 languages	-	-	80m sentences
EUbooks	✓	26 languages	-	-	173m sentences
The HOLJ Corpus	✓	english	✓	-	188 documents
Proceedings of the Old Bailey	✓	english	✓	✓	1219 documents
ParaCrawl	✓	14 languages	-	-	282m sentences

Table A.1.: Legal Corpora

B. Multi-Labeling Data Generator

Listing B.1: Data generator for the translation tasks

```
# coding=utf-8
# Copyright 2017 The Tensor2Tensor Authors.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
    implied.
# See the License for the specific language governing permissions and
# limitations under the License.

"""Data generators for summarization of jrc_acquis"""

from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

# Dependency imports

from tensor2tensor.data_generators import generator_utils
from tensor2tensor.data_generators import problem
from tensor2tensor.data_generators import text_encoder
from tensor2tensor.utils import metrics
from tensor2tensor.utils import registry

import os
import tensorflow as tf

FLAGS = tf.flags.FLAGS

EOS = text_encoder.EOS_ID

_TRAIN_DATASETS = {
    "cs":
        [
```

```
                "https://transfer.sh/uuURc/jrc_acquis.multilabeling.tar.gz",
                ("jrc_acquis.cs.documents", "jrc_acquis.cs.labels")
            ],
        "de":
            [
                "https://transfer.sh/uuURc/jrc_acquis.multilabeling.tar.gz",
                ("jrc_acquis.de.documents", "jrc_acquis.de.labels")
            ],
        "en":
            [
                "https://transfer.sh/uuURc/jrc_acquis.multilabeling.tar.gz",
                ("jrc_acquis.en.documents", "jrc_acquis.en.labels")
            ],
        "es":
            [
                "https://transfer.sh/uuURc/jrc_acquis.multilabeling.tar.gz",
                ("jrc_acquis.es.documents", "jrc_acquis.es.labels")
            ],
        "fr":
            [
                "https://transfer.sh/uuURc/jrc_acquis.multilabeling.tar.gz",
                ("jrc_acquis.fr.documents", "jrc_acquis.fr.labels")
            ],
        "it":
            [
                "https://transfer.sh/uuURc/jrc_acquis.multilabeling.tar.gz",
                ("jrc_acquis.it.documents", "jrc_acquis.it.labels")
            ],
        "sv":
            [
                "https://transfer.sh/uuURc/jrc_acquis.multilabeling.tar.gz",
                ("jrc_acquis.sv.documents", "jrc_acquis.sv.labels")
            ],
}

_TEST_DATASETS = {
        "cs":
            [
                "https://transfer.sh/uuURc/jrc_acquis.multilabeling.tar.gz",
                ("jrc_acquis.cs-test.documents", "jrc_acquis.cs-test.labels")
            ],
        "de":
            [
                "https://transfer.sh/uuURc/jrc_acquis.multilabeling.tar.gz",
                ("jrc_acquis.de-test.documents", "jrc_acquis.de-test.labels")
            ],
        "en":
            [
                "https://transfer.sh/uuURc/jrc_acquis.multilabeling.tar.gz",
                ("jrc_acquis.en-test.documents", "jrc_acquis.en-test.labels")
            ],
```

```
        "es":
            [
                "https://transfer.sh/uuURc/jrc_acquis.multilabeling.tar.gz",
                ("jrc_acquis.es-test.documents", "jrc_acquis.es-test.labels")
            ],
        "fr":
            [
                "https://transfer.sh/uuURc/jrc_acquis.multilabeling.tar.gz",
                ("jrc_acquis.fr-test.documents", "jrc_acquis.fr-test.labels")
            ],
        "it":
            [
                "https://transfer.sh/uuURc/jrc_acquis.multilabeling.tar.gz",
                ("jrc_acquis.it-test.documents", "jrc_acquis.it-test.labels")
            ],
        "sv":
            [
                "https://transfer.sh/uuURc/jrc_acquis.multilabeling.tar.gz",
                ("jrc_acquis.sv-test.documents", "jrc_acquis.sv-test.labels")
            ],

}


def download_and_extract_data(tmp_dir, dataset):
    """Download and Extract files."""
    url = dataset[0]
    print(dataset)
    compressed_filename = os.path.basename(url)
    compressed_file = generator_utils.maybe_download(
        tmp_dir, compressed_filename, url)

    for file in dataset[1]:
        tf.logging.info("Reading file: %s" % file)
        filepath = os.path.join(tmp_dir, file)

        # Extract from tar if needed.
        if not tf.gfile.Exists(filepath):
            with tarfile.open(compressed_file, "r:gz") as corpus_tar:
                corpus_tar.extractall(tmp_dir)

    documents_filename, labels_filename = dataset[1]
    documents_filepath = os.path.join(tmp_dir, documents_filename)
    labels_filepath = os.path.join(tmp_dir, labels_filename)
    return documents_filepath, labels_filepath


def token_generator(source_path, target_path, token_vocab, eos=None):
    """Generator for sequence-to-sequence tasks that uses tokens.

    This generator assumes the files at source_path and target_path have
```

```
    the same number of lines and yields dictionaries of "inputs" and "targets"
    where inputs are token ids from the " "-split source (and target, resp.) lines
    converted to integers using the token_map.

    Args:
      source_path: path to the file with source sentences.
      target_path: path to the file with target sentences.
      token_vocab: text_encoder.TextEncoder object.
      eos: integer to append at the end of each sequence (default: None).
    Yields:
      A dictionary {"inputs": source-line, "targets": target-line} where
      the lines are integer lists converted from tokens in the file lines.
    """
    eos_list = [] if eos is None else [eos]
    with tf.gfile.GFile(source_path, mode="r") as source_file:
        with tf.gfile.GFile(target_path, mode="r") as target_file:
            source, target = source_file.readline(), target_file.readline()
            while source and target:
                source_ints = token_vocab.encode(source.strip()) + eos_list
                target_ints = token_vocab.encode(target.strip()) + eos_list
                yield {"inputs": source_ints, "targets": target_ints}
                source, target = source_file.readline(), target_file.readline()


@registry.register_problem
class MultiLabelingLegal32k(problem.Text2TextProblem):
    """MultiLabeling jrc aquis docs according to their head section"""

    @property
    def is_character_level(self):
        return False

    @property
    def has_inputs(self):
        return True

    @property
    def num_shards(self):
        return 10

    @property
    def use_subword_tokenizer(self):
        return True

    @property
    def targeted_vocab_size(self):
        return 32000

    @property
    def use_train_shards_for_dev(self):
        return False
```

```
    def eval_metrics(self):
        return [
            metrics.Metrics.ACC, metrics.Metrics.ACC_TOP5,
            metrics.Metrics.ACC_PER_SEQ, metrics.Metrics.
                NEG_LOG_PERPLEXITY
        ]


@registry.register_problem
class MultiLabelingCsLegal32k(MultiLabelingLegal32k):
    """MultiLabeling cs documents"""

    @property
    def input_space_id(self):
        return problem.SpaceID.CS_TOK

    @property
    def target_space_id(self):
        return problem.SpaceID.GENERIC

    @property
    def vocab_name(self):
        return "vocab.labeling.cs"

    def generator(self, data_dir, tmp_dir, train):
        vocab = generator_utils.get_or_generate_vocab(
            data_dir, tmp_dir, self.vocab_file, self.targeted_vocab_size, [
                _TRAIN_DATASETS["cs"]])
        datasets = _TRAIN_DATASETS["cs"] if train else _TEST_DATASETS["cs"]
        document_file, labels_file = download_and_extract_data(
            tmp_dir, datasets)
        return token_generator(document_file, labels_file, vocab, EOS)


@registry.register_problem
class MultiLabelingDeLegal32k(MultiLabelingLegal32k):
    """MultiLabeling de documents"""

    @property
    def input_space_id(self):
        return problem.SpaceID.DE_TOK

    @property
    def target_space_id(self):
        return problem.SpaceID.GENERIC

    @property
    def vocab_name(self):
        return "vocab.labeling.de"
```

```
    def generator(self, data_dir, tmp_dir, train):
        vocab = generator_utils.get_or_generate_vocab(
            data_dir, tmp_dir, self.vocab_file, self.targeted_vocab_size, [
                _TRAIN_DATASETS["de"]])
        datasets = _TRAIN_DATASETS["de"] if train else _TEST_DATASETS["de"]
        document_file, labels_file = download_and_extract_data(
            tmp_dir, datasets)
        return token_generator(document_file, labels_file, vocab, EOS)


@registry.register_problem
class MultiLabelingEnLegal32k(MultiLabelingLegal32k):
    """MultiLabeling en documents"""

    @property
    def input_space_id(self):
        return problem.SpaceID.EN_TOK

    @property
    def target_space_id(self):
        return problem.SpaceID.GENERIC

    @property
    def vocab_name(self):
        return "vocab.labeling.en"

    def generator(self, data_dir, tmp_dir, train):
        vocab = generator_utils.get_or_generate_vocab(
            data_dir, tmp_dir, self.vocab_file, self.targeted_vocab_size, [
                _TRAIN_DATASETS["en"]])
        datasets = _TRAIN_DATASETS["en"] if train else _TEST_DATASETS["en"]
        document_file, labels_file = download_and_extract_data(
            tmp_dir, datasets)
        return token_generator(document_file, labels_file, vocab, EOS)


@registry.register_problem
class MultiLabelingEsLegal32k(MultiLabelingLegal32k):
    """MultiLabeling es documents"""

    @property
    def input_space_id(self):
        return problem.SpaceID.ES_TOK

    @property
    def target_space_id(self):
        return problem.SpaceID.GENERIC

    @property
    def vocab_name(self):
        return "vocab.labeling.es"
```

```
    def generator(self, data_dir, tmp_dir, train):
        vocab = generator_utils.get_or_generate_vocab(
            data_dir, tmp_dir, self.vocab_file, self.targeted_vocab_size, [
                _TRAIN_DATASETS["es"]])
        datasets = _TRAIN_DATASETS["es"] if train else _TEST_DATASETS["es"]
        document_file, labels_file = download_and_extract_data(
            tmp_dir, datasets)
        return token_generator(document_file, labels_file, vocab, EOS)


@registry.register_problem
class MultiLabelingFrLegal32k(MultiLabelingLegal32k):
    """MultiLabeling fr documents"""

    @property
    def input_space_id(self):
        return problem.SpaceID.FR_TOK

    @property
    def target_space_id(self):
        return problem.SpaceID.GENERIC

    @property
    def vocab_name(self):
        return "vocab.labeling.fr"

    def generator(self, data_dir, tmp_dir, train):
        vocab = generator_utils.get_or_generate_vocab(
            data_dir, tmp_dir, self.vocab_file, self.targeted_vocab_size, [
                _TRAIN_DATASETS["fr"]])
        datasets = _TRAIN_DATASETS["fr"] if train else _TEST_DATASETS["fr"]
        document_file, labels_file = download_and_extract_data(
            tmp_dir, datasets)
        return token_generator(document_file, labels_file, vocab, EOS)


@registry.register_problem
class MultiLabelingItLegal32k(MultiLabelingLegal32k):
    """MultiLabeling it documents"""

    @property
    def input_space_id(self):
        return problem.SpaceID.IT_TOK

    @property
    def target_space_id(self):
        return problem.SpaceID.GENERIC

    @property
    def vocab_name(self):
```

```
        return "vocab.labeling.it"

    def generator(self, data_dir, tmp_dir, train):
        vocab = generator_utils.get_or_generate_vocab(
            data_dir, tmp_dir, self.vocab_file, self.targeted_vocab_size, [
                _TRAIN_DATASETS["it"]])
        datasets = _TRAIN_DATASETS["it"] if train else _TEST_DATASETS["it"]
        document_file, labels_file = download_and_extract_data(
            tmp_dir, datasets)
        return token_generator(document_file, labels_file, vocab, EOS)


@registry.register_problem
class MultiLabelingSvLegal32k(MultiLabelingLegal32k):
    """MultiLabeling sv documents"""

    @property
    def input_space_id(self):
        return problem.SpaceID.SV_TOK

    @property
    def target_space_id(self):
        return problem.SpaceID.GENERIC

    @property
    def vocab_name(self):
        return "vocab.labeling.sv"

    def generator(self, data_dir, tmp_dir, train):
        vocab = generator_utils.get_or_generate_vocab(
            data_dir, tmp_dir, self.vocab_file, self.targeted_vocab_size, [
                _TRAIN_DATASETS["sv"]])
        datasets = _TRAIN_DATASETS["sv"] if train else _TEST_DATASETS["sv"]
        document_file, labels_file = download_and_extract_data(
            tmp_dir, datasets)
        return token_generator(document_file, labels_file, vocab, EOS)
```

List of Figures

List of Tables

Glossary

Acquis Communautaire A collection of documents which encompasses all rights and responsibilities that are obligatory to the European member states.

Backpropagation Algorithm A supervised learning algorithm for propagating back updates of parameters according to the result of an error function.

Computational Block An encapsulated construct of operations that is integrated as part of an artificial neural network.

Corpus A collection of written text.

Deep Learning A machine learning discipline involving the usage of deep artificial neural networks..

EuroVoc Thesaurus A collection of over 6000 hierarchically determined classes covering the areas of operation of the European Union.

Graphical Processor Unit A processor specialized and optimized in computing graphics for computer games and simulations.

Hyperparameter A customizable value that describes a property of an artificial neural network.

Legal Domain The entirety of matters and activities in regard to the law.

Medium A resource used for the communication which carries information.

Parameter A trainable value of an artificial neural network.

Softmax Function A mathematical function mapping a vector v with D dimensions to a vector z with D dimensions in which all components lie in the value range $(0, 1)$ and sum up to 1.

State-of-the-Art The latest state of development.

Supervised Learning A machine learning technique which uses a training set of labeled examples.

Tensorflow A machine learning library used for building, training and evaluating artificial neural networks.

TFRecord A binary format for efficiently storing data used in Tensorflow.

Tokenization Process of demarcating and possibly classifying sections of a string of input characters.

Transfer Learning Sharing knowledge gained while solving one problem and applying it to a different but related problem.

Bibliography

[1] N. Hajlaoui, D. Kolovratnik, J. Väyrynen, R. Steinberger, and D. Varga, "Dcep-digital corpus of the european parliament.," in *LREC*, 2014, pp. 3164–3171.

[2] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *MT summit*, vol. 5, 2005, pp. 79–86.

[3] R. Steinberger, B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, D. Tufis, and D. Varga, "The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages," *CoRR*, vol. abs/cs/0609058, 2006. arXiv: cs/0609058.

[4] L. Kaiser, A. N. Gomez, N. Shazeer, A. Vaswani, N. Parmar, L. Jones, and J. Uszkoreit, "One model to learn them all," *CoRR*, vol. abs/1706.05137, 2017. arXiv: 1706.05137.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. arXiv: 1706.03762.

[6] R. Steinberger, M. Ebrahim, and M. Turchi, "JRC eurovoc indexer JEX - A freely available multi-label categorisation tool," *CoRR*, vol. abs/1309.5223, 2013. arXiv: 1309.5223.

[7] B. Widrow, "Generalization and information storage in networks of adaline "neurons"," in *Self-Organizing Systems 1962*, M. Yovits, G. Jacobi, and G. Goldstein, Eds., Chicago 1962: Spartan, Washington, 1962, pp. 435–461.

[8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, p. 533, 1986.

[9] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," in *Shape, contour and grouping in computer vision*, Springer, 1999, pp. 319–345.

[10] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, "Cudnn: Efficient primitives for deep learning," *CoRR*, vol. abs/1410.0759, 2014. arXiv: 1410.0759.

[11] P. Koehn, A. Birch, and R. Steinberger, "462 machine translation systems for europe," *Proceedings of MT Summit XII*, pp. 65–72, 2009.

[12] O.-M. Sulea, M. Zampieri, S. Malmasi, M. Vela, L. P. Dinu, and J. van Genabith, "Exploring the use of text classification in the legal domain," *ArXiv preprint arXiv:1710.09306*, 2017.

[13] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale deep unsupervised learning using graphics processors," in *Proceedings of the 26th annual international conference on machine learning*, ACM, 2009, pp. 873–880.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 2009, pp. 248–255.

[15] S. Spiegler, "Statistcs of the common crawl corpus 2012," Technical report, SwiftKey, Tech. Rep., 2013.

[16] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, IEEE, 2015, pp. 5206–5210.

[17] A. Vaswani, S. Bengio, E. Brevdo, F. Chollet, A. N. Gomez, S. Gouws, L. Jones, Ł. Kaiser, N. Kalchbrenner, N. Parmar, *et al.*, "Tensor2tensor for neural machine translation," *ArXiv preprint arXiv:1803.07416*, 2018.

[18] S. R. Anderson, "How many languages are there in the world," *Linguistic Society of America*, 2004.

[19] M. Harvey, "What's so special about legal translation?" *Meta: Journal des traducteurs/Meta: Translators' Journal*, vol. 47, no. 2, pp. 177–185, 2002.

[20] G. Garzone, "Legal translation and functionalist approaches: A contradiction in terms," *ASTTI/ETI*, vol. 395, 2000.

[21] G. Murray, S. Renals, and J. Carletta, "Extractive summarization of meeting recordings.," 2005.

[22] Y.-F. Pan, X. Hou, and C.-L. Liu, "Text localization in natural scene images based on conditional random field," in *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, IEEE, 2009, pp. 6–10.

[23] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining.," in *LREc*, vol. 10, 2010.

[24] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Identifying suspicious urls: An application of large-scale online learning," in *Proceedings of the 26th annual international conference on machine learning*, ACM, 2009, pp. 681–688.

[25] A. Ritter, S. Clark, O. Etzioni, *et al.*, "Named entity recognition in tweets: An experimental study," in *Proceedings of the conference on empirical methods in natural language processing*, Association for Computational Linguistics, 2011, pp. 1524–1534.

[26] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1700–1709.

[27] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[28] J. Read, B. Pfahringer, and G. Holmes, "Multi-label classification using ensembles of pruned sets," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, IEEE, 2008, pp. 995–1000.

[29] N. Ghamrawi and A. McCallum, "Collective multi-label classification," in *Proceedings of the 14th ACM international conference on Information and knowledge management*, ACM, 2005, pp. 195–200.

[30] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*, 1990, pp. 396–404.

[31] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," *ArXiv preprint arXiv:1312.6026*, 2013.

[32] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, IEEE, 2013, pp. 6645–6649.

[33] R. Caruana and V. R. de Sa, "Promoting poor features to supervisors: Some inputs work better as outputs," in *Advances in Neural Information Processing Systems 9*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds., MIT Press, 1997, pp. 389–395.

[34] Y. S. Abu-Mostafa, "Learning from hints in neural networks," *Journal of complexity*, vol. 6, no. 2, pp. 192–198, 1990.

[35] R. Caruna, "Multitask learning: A knowledge-based source of inductive bias," in *Machine Learning: Proceedings of the Tenth International Conference*, 1993, pp. 41–48.

[36] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, Jul. 1997, ISSN: 0885-6125. DOI: 10.1023/A:1007379606734.

[37] K. Wolk and K. Marasek, "Building subject-aligned comparable corpora and mining it for truly parallel sentence pairs," *CoRR*, vol. abs/1509.08881, 2015. arXiv: 1509.08881.

[38] R. Steinberger, A. Eisele, S. Klocek, S. Pilos, and P. Schlüter, "DGT-TM: A freely available translation memory in 22 languages," *CoRR*, vol. abs/1309.5226, 2013. arXiv: 1309.5226.

[39] J. Tiedemann, "Parallel data, tools and interfaces in opus.," in *LREC*, vol. 2012, 2012, pp. 2214–2218.

[40] C. Grover, B. Hachey, and I. Hughson, "The holj corpus. supporting summarisation of legal texts," in *Proceedings of the 5th International Workshop on Linguistically Interpreted Corpora*, 2004.

[41] K. M. Hermann, T. Kociský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," *CoRR*, vol. abs/1506.03340, 2015. arXiv: 1506.03340.

[42] R. Nallapati, B. Xiang, and B. Zhou, "Sequence-to-sequence rnns for text summarization," *CoRR*, vol. abs/1602.06023, 2016. arXiv: 1602.06023.

[43] C. Napoles, M. Gormley, and B. Van Durme, "Annotated gigaword," in *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, ser. AKBC-WEKEX '12, Montreal, Canada: Association for Computational Linguistics, 2012, pp. 95–100.

[44] P. Over, H. Dang, and D. Harman, "Duc in context," *Information Processing & Management*, vol. 43, no. 6, pp. 1506–1520, 2007.

[45] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," *CoRR*, vol. abs/1509.00685, 2015. arXiv: 1509.00685.

[46] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," *CoRR*, vol. abs/1705.04304, 2017. arXiv: 1705.04304.

[47] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," *Text Summarization Branches Out*, 2004.

[48] M. Huber, "The old bailey proceedings, 1674-1834. evaluating and annotating a corpus of 18th-and 19th-century spoken english," *Annotating variation and change*, 2007.

[49] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*, Association for Computational Linguistics, 2002, pp. 311–318.

[50] K. Ahmed, N. S. Keskar, and R. Socher, "Weighted transformer network for machine translation," *CoRR*, vol. abs/1711.02132, 2017. arXiv: 1711.02132.

[51] J. Gu, J. Bradbury, C. Xiong, V. O. K. Li, and R. Socher, "Non-autoregressive neural machine translation," *CoRR*, vol. abs/1711.02281, 2017. arXiv: 1711.02281.

[52] H. Sajjad, N. Durrani, F. Dalvi, Y. Belinkov, and S. Vogel, "Neural machine translation training in a multi-domain scenario," *CoRR*, vol. abs/1708.08712, 2017. arXiv: 1708.08712.

[53] W. Zeng, W. Luo, S. Fidler, and R. Urtasun, "Efficient summarization with read-again and copy mechanism," *CoRR*, vol. abs/1611.03382, 2016. arXiv: 1611.03382.

[54] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," *CoRR*, vol. abs/1704.04368, 2017. arXiv: 1704.04368.

[55] L. Liu, Y. Lu, M. Yang, Q. Qu, J. Zhu, and H. Li, "Generative adversarial network for abstractive text summarization," *CoRR*, vol. abs/1711.09357, 2017. arXiv: 1711.09357.

[56] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2014, pp. 2672–2680.

[57] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer, "Generating wikipedia by summarizing long sequences," *CoRR*, vol. abs/1801.10198, 2018. arXiv: 1801.10198.

[58] J. Hasselqvist, N. Helmertz, and M. Kågebäck, "Query-based abstractive summarization using neural networks," *CoRR*, vol. abs/1712.06100, 2017. arXiv: 1712.06100.

[59] B. Hachey and C. Grover, "Automatic legal text summarisation: Experiments with summary structuring," in *Proceedings of the 10th international conference on Artificial intelligence and law*, ACM, 2005, pp. 75–84.

[60] B. Pouliquen, R. Steinberger, and C. Ignat, "Automatic annotation of multilingual text collections with a conceptual thesaurus," *CoRR*, vol. abs/cs/0609059, 2006. arXiv: cs/0609059.

[61] K. Sarinnapakorn and M. Kubat, "Combining subclassifiers in text categorization: A dst-based solution and a case study," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 12, pp. 1638–1651, 2007.

[62] E. L. Mencía and J. Fúrnkranz, "Efficient multilabel classification algorithms for large-scale problems in the legal domain," in *Semantic Processing of Legal Texts*, Springer, 2010, pp. 192–215.

[63] G. Boella, L. Di Caro, D. Rispoli, and L. Robaldo, *A system for classifying multi-label text into eurovoc*, Jun. 2013.

[64] W. Alschner and D. Skougarevskiy, "Towards an automated production of legal texts using recurrent neural networks," 2017.

[65] A. Wyner and G. Casini, "A deep learning approach to contract element extraction," *Legal Knowledge and Information Systems*, p. 155, 2017.

[66] D. L. Chen and J. Eagel, "Can machine learning help predict the outcome of asylum adjudications?" In *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*, ACM, 2017, pp. 237–240.

[67] I. NEJADGHOLI, R. BOUGUENG, and S. WITHERSPOON, "A semi-supervised training method for semantic search of legal facts in canadian immigration cases," *Legal Knowledge and Information Systems*, p. 125, 2017.

[68] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, ACM, 2008, pp. 160–167.

[69] X. Liu, J. Gao, X. He, L. Deng, K. Duh, and Y.-Y. Wang, "Representation learning using multi-task deep neural networks for semantic classification and information retrieval," 2015.

[70] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," *ArXiv preprint arXiv:1605.05101*, 2016.

[71] ——, "Deep multi-task learning with shared memory," *ArXiv preprint arXiv:1609.07222*, 2016.

[72] H. Zhang, L. Xiao, Y. Wang, and Y. Jin, "A generalized recurrent neural architecture for text classification with multi-task learning," *ArXiv preprint arXiv:1707.02892*, 2017.

[73] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR*, vol. abs/1406.1078, 2014. arXiv: 1406.1078.

[74] P. Koehn, *Statistical machine translation*. Cambridge University Press, 2009.

[75] M. Ehrmann, M. Turchi, and R. Steinberger, "Building a multilingual named entity-annotated corpus using annotation projection," in *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, 2011, pp. 118–124.

[76] J. Steinberger, P. Lenkova, M. Kabadjov, R. Steinberger, and E. Van der Goot, "Multilingual entity-centered sentiment analysis evaluated by parallel corpora," in *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, 2011, pp. 770–775.

[77] D. Varga, P. Halácsy, A. Kornai, V. Nagy, L. Németh, and V. Trón, "Parallel corpora for medium density languages," *AMSTERDAM STUDIES IN THE THEORY AND HISTORY OF LINGUISTIC SCIENCE SERIES 4*, vol. 292, p. 247, 2007.

[78] R. Steinberger, M. Ebrahim, A. Poulis, M. Carrasco-Benitez, P. Schlüter, M. Przybyszewski, and S. Gilbro, "An overview of the european union's highly multilingual parallel corpora," *Language resources and evaluation*, vol. 48, no. 4, pp. 679–707, 2014.

[79] R. Skadiņš, J. Tiedemann, R. Rozis, and D. Deksne, "Billions of parallel words for free: Building and using the eu bookshop corpus," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2014, pp. 1850–1855.

[80] H. Grabbe, "European union conditionality and the acquis communautaire," *International political science review*, vol. 23, no. 3, pp. 249–268, 2002.

[81] T. Erjavec, "Text encoding initiative guidelines and their localisation.," *INFOtheca-Journal of Informatics & Librarianship*, vol. 11, no. 1, 2010.

[82] W. A. Gale and K. W. Church, "A program for aligning sentences in bilingual corpora," *Computational linguistics*, vol. 19, no. 1, pp. 75–102, 1993.

[83] E. G. P.-S. Luquet, "Multilingual lexical database generation from parallel texts with endogenous resources," 2005.

[84] S. Teufel and M. Moens, "Summarizing scientific articles: Experiments with relevance and rhetorical status," *Computational linguistics*, vol. 28, no. 4, pp. 409–445, 2002.

[85] C. Grover, C. Matheson, A. Mikheev, and M. Moens, "Lt ttt-a flexible tokenisation tool," in *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, 2000.

[86] G. Minnen, J. Carroll, and D. Pearce, "Robust, applied morphological generation," in *Proceedings of the first international conference on Natural language generation-Volume 14*, Association for Computational Linguistics, 2000, pp. 201–208.

[87] J. R. Curran and S. Clark, "Language independent ner using a maximum entropy tagger," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, Association for Computational Linguistics, 2003, pp. 164–167.

[88] A. Eisele and Y. Chen, "Multiun: A multilingual corpus from united nation documents," in *Proceedings of the Seventh conference on International Language Resources and Evaluation*, D. Tapias, M. Rosner, S. Piperidis, J. Odjik, J. Mariani, B. Maegaard, K. Choukri, and N. C. (Chair), Eds., European Language Resources Association (ELRA), May 2010, pp. 2868–2872.

[89] S. Bird and E. Loper, "Nltk: The natural language toolkit," in *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, Association for Computational Linguistics, 2004, p. 31.

[90] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Y. Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng,

Tensorflow: Large-scale machine learning on heterogeneous systems, Software available from tensorflow.org, 2015.

[91] R. Collobert, S. Bengio, and J. Mariéthoz, "Torch: A modular machine learning software library," Idiap, Tech. Rep., 2002.

[92] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS workshop*, 2011.

[93] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: A cpu and gpu math compiler in python," in *Proc. 9th Python in Science Conf*, vol. 1, 2010.

[94] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *CoRR*, vol. abs/1408.5093, 2014. arXiv: 1408.5093.

[95] D. Team, "Deeplearning4j: Open-source distributed deep learning for the jvm," *Apache Software Foundation License*, vol. 2, 2016.

[96] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, Association for Computational Linguistics, 2007, pp. 177–180.

[97] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. van den Oord, A. Graves, and K. Kavukcuoglu, "Neural machine translation in linear time," *CoRR*, vol. abs/1610.10099, 2016. arXiv: 1610.10099.

[98] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016. arXiv: 1609.03499.

[99] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. V. Le, G. E. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *CoRR*, vol. abs/1701.06538, 2017. arXiv: 1701.06538.

[100] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *CoRR*, vol. abs/1610.02357, 2016. arXiv: 1610.02357.

[101] L. Kaiser, A. N. Gomez, and F. Chollet, "Depthwise separable convolutions for neural machine translation," *CoRR*, vol. abs/1706.03059, 2017. arXiv: 1706.03059.

[102] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *CoRR*, vol. abs/1508.07909, 2015. arXiv: 1508.07909.

[103] M. Popović, "Chrf: Character n-gram f-score for automatic mt evaluation," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*, 2015, pp. 392–395.

[104] F. Hieber, T. Domhan, M. Denkowski, D. Vilar, A. Sokolov, A. Clifton, and M. Post, "Sockeye: a toolkit for neural machine translation," *ArXiv preprint arXiv:1712.05690*, Dec. 2017. arXiv: 1712.05690 [cs.CL].

[105] G. Durrett, T. Berg-Kirkpatrick, and D. Klein, "Learning-based single-document summarization with compression and anaphoricity constraints," *CoRR*, vol. abs/1603.08887, 2016. arXiv: 1603.08887.