# Structural and Topical Dimensions in Multi-Task Patent Translation

**Katharina Wäschle** and **Stefan Riezler**
Department of Computational Linguistics
Heidelberg University, Germany
{waeschle,riezler}@cl.uni-heidelberg.de

## Abstract

Patent translation is a complex problem due to the highly specialized technical vocabulary and the peculiar textual structure of patent documents. In this paper we analyze patents along the orthogonal dimensions of topic and textual structure. We view different patent classes and different patent text sections such as title, abstract, and claims, as separate translation tasks, and investigate the influence of such tasks on machine translation performance. We study multi-task learning techniques that exploit commonalities between tasks by mixtures of translation models or by multi-task meta-parameter tuning. We find small but significant gains over task-specific training by techniques that model commonalities through shared parameters. A by-product of our work is a parallel patent corpus of 23 million German-English sentence pairs.

## 1 Introduction

Patents are an important tool for the protection of intellectual property and also play a significant role in business strategies in modern economies. Patent translation is an enabling technique for patent prior art search which aims to detect a patent's novelty and thus needs to be cross-lingual for a multitude of languages. Patent translation is complicated by a highly specialized vocabulary, consisting of technical terms specific to the field of invention the patent relates to. Patents are written in a sophisticated legal jargon ("patentese") that is not found in everyday language and exhibits a complex textual structure. Also, patents are often intentionally ambiguous or vague in order to maximize the coverage of the claims.

In this paper, we analyze patents with respect to the orthogonal dimensions of topic – the technical field covered by the patent – and structure – a patent's text sections –, with respect to their influence on machine translation performance.

The topical dimension of patents is characterized by the International Patent Classification (IPC)[1] which categorizes patents hierarchically into 8 sections, 120 classes, 600 subclasses, down to 70,000 subgroups at the leaf level. Table 1 shows the 8 top level sections.

| | |
|---|---|
| A | Human Necessities |
| B | Performing Operations, Transporting |
| C | Chemistry, Metallurgy |
| D | Textiles, Paper |
| E | Fixed Constructions |
| F | Mechanical Engineering, Lighting, Heating, Weapons |
| G | Physics |
| H | Electricity |

Table 1: IPC top level sections.

Orthogonal to the patent classification, patent documents can be sub-categorized along the dimension of textual structure. Article 78.1 of the European Patent Convention (EPC) lists all sections required in a patent document[2]:

> "A European patent application shall contain:
>
> (a) a **request for the grant** of a European patent;

[1] http://www.wipo.int/classifications/ipc/en/

[2] Highlights by the authors.

(b) a **description** of the invention;
(c) one or more **claims**;
(d) any **drawings** referred to in the description or the claims;
(e) an **abstract**,

and satisfy the requirements laid down in the Implementing Regulations."

The request for grant contains the patent title; thus a patent document comprises the textual elements of title, description, claim, and abstract.

We investigate whether it is worthwhile to treat different values along the structural and topical dimensions as different tasks that are not completely independent of each other but share some commonalities, yet differ enough to counter a simple pooling of data. For example, we consider different tasks such as patents from different IPC classes, or along an orthogonal dimension, patent documents of all IPC classes but consisting only of titles or only of claims. We ask whether such tasks should be addressed as separate translation tasks, or whether translation performance can be improved by learning several tasks simultaneously through shared models that are more sophisticated than simple data pooling. Our goal is to learn a patent translation system that performs well across several different tasks, thus benefits from shared information, but is yet able to address the specifics of each task.

One contribution of this paper is a thorough analysis of the differences and similarities of multilingual patent data along the dimensions of textual structure and topic. The second contribution is the experimental investigation of the influence of various such tasks on patent translation performance. Starting from baseline models that are trained on individual tasks or on data pooled from all tasks, we apply mixtures of translation models and multi-task minimum error rate training to multiple patent translation tasks. A by-product of our research is a parallel patent corpus of over 23 million sentence pairs.

## 2 Related work

Multi-task learning has mostly been discussed under the name of multi-domain adaptation in the area of statistical machine translation (SMT). If we consider domains as tasks, domain adaptation is a special two-task case of multi-task learning. Most previous work has concentrated on adapting unsupervised generative modules such as translation models or language models to new tasks. For example, transductive approaches have used automatic translations of monolingual corpora for self-training modules of the generative SMT pipeline (Ueffing et al., 2007; Schwenk, 2008; Bertoldi and Federico, 2009). Other approaches have extracted parallel data from similar or comparable corpora (Zhao et al., 2004; Snover et al., 2008). Several approaches have been presented that train separate translation and language models on task-specific subsets of the data and combine them in different mixture models (Foster and Kuhn, 2007; Koehn and Schroeder, 2007; Foster et al., 2010). The latter kind of approach is applied in our work to multiple patent tasks.

Multi-task learning efforts in patent translation have so far been restricted to experimental combinations of translation and language models from different sets of IPC sections. For example, Utiyama and Isahara (2007) and Tinsley et al. (2010) investigate translation and language models trained on different sets of patent sections, with larger pools of parallel data improving results. Ceauşu et al. (2011) find that language models always and translation model mostly benefit from larger pools of data from different sections. Models trained on pooled patent data are used as baselines in our approach.

The machine learning community has developed several different formalizations of the central idea of trading off optimality of parameter vectors for each task-specific model and closeness of these model parameters to the average parameter vector across models. For example, starting from a separate SVM for each task, Evgeniou and Pontil (2004) present a regularization method that trades off optimization of the task-specific parameter vectors and the distance of each SVM to the average SVM. Equivalent formalizations replace parameter regularization by Bayesian prior distributions on the parameters (Finkel and Manning, 2009) or by augmentation of the feature space with domain independent features (Daumé, 2007). Besides SVMs, several learning algorithms have been extended to the multi-task scenario in a parameter regularization setting, e.g., perceptron-type algorithms (Dredze et al., 2010) or boosting (Chapelle et al., 2011). Further variants include different formalizations of norms for parameter regularization, e.g., $\ell_{1,2}$ regularization

(Obozinski et al., 2010) or $\ell_{1,\infty}$ regularization (Quattoni et al., 2009), where only the features that are most important across all tasks are kept in the model. In our experiments, we apply parameter regularization for multi-task learning to minimum error rate training for patent translation.

## 3 Extraction of a parallel patent corpus from comparable data

Our work on patent translation is based on the MAREC[3] patent data corpus. MAREC contains over 19 million patent applications and granted patents in a standardized format from four patent organizations (European Patent Office (EP), World Intellectual Property Organization (WO), United States Patent and Trademark Office (US), Japan Patent Office (JP)), from 1976 to 2008. The data for our experiments are extracted from the EP and WO collections which contain patent documents that include translations of some of the patent text. To extract such parallel patent sections, we first determine the longest instance, if different kinds[4] exist for a patent. We assume titles to be sentence-aligned by default, and define sections with a token ratio larger than 0.7 as parallel. For the language pair German-English we extracted a total of 2,101,107 parallel titles, 291,716 parallel abstracts, and 735,667 parallel claims sections.

The lack of directly translated descriptions poses a serious limitation for patent translation, since this section constitutes the largest part of the document. It is possible to obtain comparable descriptions from related patents that have been filed in different countries and are connected through the patent family id. We extracted 172,472 patents that were both filed with the USPTO and the EPO and contain an English and a German description, respectively.

For sentence alignment, we used the Gargantua[5] tool (Braune and Fraser, 2010) that filters a sentence-length based alignment with IBM Model-1 lexical word translation probabilities, estimated on parallel data obtained from the first-pass alignment. This yields the parallel corpus listed in table 2 with high input-output ratios for claims, and much lower ratios for abstracts and descriptions, showing that claims exhibit a natural parallelism due to their structure, while abstracts and descriptions are considerably less parallel. Removing duplicates and adding parallel titles results in a corpus of over 23 million parallel sentence pairs.

| | output | de ratio | en ratio |
|---|---|---|---|
| abstract | 720,571 | 92.36% | 76.81% |
| claims | 8,346,863 | 97.82% | 96.17% |
| descr. | 14,082,381 | 86.23% | 82.67% |

Table 2: Number of parallel sentences in output with input/output ratio of sentence aligner.

Differences between the text sections become visible in an analysis of token to type ratios. Table 3 gives the average number of tokens compared to the average type frequencies for a window of 100,000 tokens from every subsection. It shows that titles contain considerably fewer tokens than other sections, however, the disadvantage is partially made up by a relatively large amount of types, indicated by a lower average type frequency.

| | tokens | | types | |
|---|---|---|---|---|
| | de | en | de | en |
| title | 6.5 | 8.0 | 2.9 | 4.8 |
| abstract | 37.4 | 43.2 | 4.3 | 9.0 |
| claims | 53.2 | 61.3 | 5.5 | 9.5 |
| description | 27.5 | 35.5 | 4.0 | 7.0 |

Table 3: Average number of tokens and average type frequencies in text sections.

We reserved patent data published between 1979 and 2007 for training and documents published in 2008 for tuning and testing in SMT. For the dimension of text sections, we sampled 500,000 sentences – distributed across all IPC sections – for training and 2,000 sentences for each text section for development and testing. Because of a relatively high number of identical sentences in test and training set for titles, we removed the overlap for this section.

Table 4 shows the distribution of IPC sections on claims, with the smallest class accounting for

[3] http://www.ir-facility.org/prototypes/marec

[4] A patent kind code indicates the document stage in the filing process, e.g., A for applications and B for granted patents, with publication levels from 1-9. See http://www.wipo.int/standards/en/part_03.html.

[5] http://gargantua.sourceforge.net

around 300,000 parallel sentences. In order to obtain similar amounts of training data for each task along the topical dimension, we sampled 300,000 sentences from each IPC class for training, and 2,000 sentences for each IPC class for development and testing.

| | |
|---|---|
| A | 1,947,542 |
| B | 2,522,995 |
| C | 2,263,375 |
| D | 299,742 |
| E | 353,910 |
| F | 1,012,808 |
| G | 2,066,132 |
| H | 1,754,573 |

Table 4: Distribution of IPC sections on claims.

## 4 Machine translation experiments

### 4.1 Individual task baselines

For our experiments we used the phrase-based, open-source SMT toolkit Moses[6] (Koehn et al., 2007). For language modeling, we computed 5-gram models using IRSTLM[7] (Federico et al., 2008) and queried the model with KenLM (Heafield, 2011). BLEU (Papineni et al., 2001) scores were computed up to 4-grams on lowercased data.

| | Europarl-v6 | | MAREC | |
|---|---|---|---|---|
| | BLEU | OOV | BLEU | OOV |
| abstract | 0.1726 | 14.40% | **0.3721** | 3.00% |
| claim | 0.2301 | 15.80% | **0.4711** | 4.20% |
| title | 0.0964 | 26.00% | **0.3228** | 9.20% |

Table 5: BLEU scores and OOV rate for Europarl baseline and MAREC model.

Table 5 shows a first comparison of results of Moses models trained on 500,000 parallel sentences from patent text sections balanced over IPC classes, against Moses trained on 1.7 Million sentences of parliament proceedings from Europarl[8] (Koehn, 2005). The best result on each section is indicated in **bold** face. The Europarl model performs very poorly on all three sections in comparison to the task-specific MAREC model, although the former has been learned on more than three times the amount of data. An analysis of the output of both system shows that the Europarl model suffers from two problems: Firstly, there is an obvious out of vocabulary (OOV) problem of the Europarl model compared to the MAREC model. Secondly, the Europarl model suffers from incorrect word sense disambiguation, as illustrated by the samples in table 6.

| source | steuerbar | leitet |
|---|---|---|
| Europarl | taxable | is in charge of |
| MAREC | controllable | guiding |
| reference | controllable | guides |

Table 6: Output of Europarl model on MAREC data.

Table 7 shows the results of the evaluation across text sections; we measured the performance of separately trained and tuned individual models on every section. The results allow some conclusions about the textual characteristics of the sections and indicate similarities. Naturally, every task is best translated with a model trained on the respective section, as the BLEU scores on the diagonal are the highest in every column. Accordingly, we are interested in the runner-up on each section, which is indicated in **bold** font. The results on abstracts suggest that this section bears the strongest resemblance to claims, since the model trained on claims achieves a respectable score. The abstract model seems to be the most robust and varied model, yielding the runner-up score on all other sections. Claims are easiest to translate, yielding the highest overall BLEU score of 0.4879. In contrast to that, all models score considerably lower on titles.

| | test | | | |
|---|---|---|---|---|
| train | abstract | claim | title | desc. |
| abstract | *0.3737* | **0.4076** | **0.2681** | **0.2812** |
| claim | **0.3416** | *0.4879* | 0.2420 | 0.2623 |
| title | 0.2839 | 0.3512 | *0.3196* | 0.1743 |
| desc. | 0.32189 | **0.403** | 0.2342 | *0.3347* |

Table 7: BLEU scores for 500k individual text section models.

The cross-section evaluation on the IPC classes (table 8) shows similar patterns. Each section

[6] http://statmt.org/moses/

[7] http://sourceforge.net/projects/irstlm/

[8] http://www.statmt.org/europarl/

is best translated with a model trained on data from the same section. Note that best section scores vary considerably, ranging from 0.5719 on C to 0.4714 on H, indicating that higher-scoring classes, such as C and A, are more homogeneous and therefore easier to translate. C, the Chemistry section, presumably benefits from the fact that the data contain chemical formulae, which are language-independent and do not have to be translated. Again, for determining the relationship between the classes, we examine the best runner-up on each section, considering the BLEU score, although asymmetrical, as a kind of measure of similarity between classes. We can establish symmetric relationships between sections A and C, B and F as well as G and H, which means that the models are mutual runner-up on the other's test section.

The similarities of translation tasks established in the previous section can be confirmed by information-theoretic similarity measures that perform a pairwise comparison of the vocabulary probability distribution of each task-specific corpus. This distribution is calculated on the basis of the 500 most frequent words in the union of two corpora, normalized by vocabulary size. As metric we use the $\mathcal{A}$-distance measure of Kifer et al. (2004). If $\mathcal{A}$ is the set of events on which the word distributions of two corpora are defined, then the $\mathcal{A}$-distance is the supremum of the difference of probabilities assigned to the same event. Low distance means higher similarity.

Table 9 shows the $\mathcal{A}$-distance of corpora specific to IPC classes. The most similar section or sections – apart from the section itself on the diagonal – is indicated in **bold** face. The pairwise similarity of A and C, B and F, G and H obtained by BLEU score is confirmed. Furthermore, a close similarity between E and F is indicated. G and H (electricity and physics, respectively) are very similar to each other but not close to any other section apart from B.

### 4.2 Task pooling and mixture

One straightforward technique to exploit commonalities between tasks is pooling data from separate tasks into a single training set. Instead of a trivial enlargement of training data by pooling, we train the pooled models on the same amount of sentences as the individual models. For instance, the pooled model for the pairing of IPC section B and C is trained on a data set composed of 150,000 sentences from each IPC section. The pooled model for pairing data from abstracts and claims is trained on data composed of 250,000 sentences from each text section.

Another approach to exploit commonalities between tasks is to train separate language and translation models[9] on the sentences from each task and combine the models in the global log-linear model of the SMT framework, following Foster and Kuhn (2007) and Koehn and Schroeder (2007). Model combination is accomplished by adding additional language model and translation model features to the log-linear model and tuning the additional meta-parameters by standard minimum error rate training (Bertoldi et al., 2009).

We try out mixture and pooling for all pairwise combinations of the three structural sections, for which we have high-quality data, i.e. abstract, claims and title. Due to the large number of possible combinations of IPC sections, we limit the experiments to pairs of similar sections, based on the $\mathcal{A}$-distance measure.

Table 10 lists the results for two combinations of data from different sections: a log-linear mixture of separately trained models and simple pooling, i.e. concatenation, of the training data. Overall, the mixture models perform slightly better than the pooled models on the text sections, although the difference is significant only in two cases. This is indicated by highlighting best results in **bold** face (with more than one result highlighted if the difference is not significant).[10]

We investigate the same mixture and pooling techniques on the IPC sections we considered pairwise similar (see table 11). Somehow contradicting the former results, the mixture models perform significantly worse than the pooled model on three sections. This might be the result of inadequate tuning, since most of the time the MERT algorithm did not converge after the maximum number of iterations, due to the larger number of features when using several models.

[9] Following Duh et al. (2010), we use the alignment model trained on the pooled data set in the phrase extraction phase of the separate models. Similarly, we use a globally trained lexical reordering model.

[10] For assessing significance, we apply the approximate randomization method described in Riezler and Maxwell (2005). We consider pairwise differing results scoring a p-value smaller than 0.05 as significant; the assessment is repeated three times and the average value is taken.

| | test | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| train | A | B | C | D | E | F | G | H |
| A | *0.5349* | 0.4475 | **0.5472** | 0.4746 | 0.4438 | 0.4523 | 0.4318 | 0.4109 |
| B | 0.4846 | *0.4736* | 0.5161 | **0.4847** | **0.4578** | **0.4734** | 0.4396 | 0.4248 |
| C | **0.5047** | 0.4257 | *0.5719* | 0.462 | 0.4134 | 0.4249 | 0.409 | 0.3845 |
| D | 0.47 | 0.4387 | 0.5106 | *0.5167* | 0.4344 | 0.4435 | 0.407 | 0.3917 |
| E | 0.4486 | 0.4458 | 0.4681 | 0.4531 | *0.4771* | 0.4591 | 0.4073 | 0.4028 |
| F | 0.4595 | **0.4588** | 0.4761 | 0.4655 | 0.4517 | *0.4909* | 0.422 | 0.4188 |
| G | 0.4935 | 0.4489 | 0.5239 | 0.4629 | 0.4414 | 0.4565 | *0.4748* | **0.4532** |
| H | 0.4628 | 0.4484 | 0.4914 | 0.4621 | 0.4421 | 0.4616 | **0.4588** | *0.4714* |

Table 8: BLEU scores for 300k individual IPC section models.

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| A | *0* | 0.1303 | **0.1317** | 0.1311 | 0.188 | 0.186 | 0.164 | 0.1906 |
| B | **0.1302** | *0* | 0.2388 | **0.1242** | **0.0974** | **0.0875** | 0.1417 | 0.1514 |
| C | **0.1317** | 0.2388 | *0* | 0.1992 | 0.311 | 0.3068 | 0.2506 | 0.2825 |
| D | **0.1311** | 0.1242 | 0.1992 | *0* | 0.1811 | 0.1808 | 0.1876 | 0.201 |
| E | 0.188 | 0.0974 | 0.311 | 0.1811 | *0* | **0.0921** | 0.2058 | 0.2025 |
| F | 0.186 | **0.0875** | 0.3068 | 0.1808 | **0.0921** | *0* | 0.1824 | 0.1743 |
| G | 0.164 | 0.1417 | 0.2506 | 0.1876 | 0.2056 | 0.1824 | *0* | **0.064** |
| H | 0.1906 | 0.1514 | 0.2825 | 0.201 | 0.2025 | 0.1743 | **0.064** | *0* |

Table 9: Pairwise $\mathcal{A}$-distance for 300k IPC training sets.

| train | test | pooling | mixture |
|---|---|---|---|
| abstract-claim | abstract | **0.3703** | **0.3704** |
| | claim | 0.4809 | **0.4834** |
| claim-title | claim | **0.4799** | **0.4789** |
| | title | **0.3269** | **0.328** |
| title-abstract | title | **0.3311** | **0.3275** |
| | abstract | 0.3643 | **0.366** |

Table 10: Mixture and pooling on text sections.

| train | test | pooling | mixture |
|---|---|---|---|
| A-C | A | **0.5271** | **0.5274** |
| | C | **0.5664** | 0.5632 |
| B-F | B | **0.4696** | 0.4354 |
| | F | **0.4859** | 0.4769 |
| G-H | G | **0.4735** | **0.4754** |
| | H | 0.4634 | **0.467** |

Table 11: Mixture and pooling on IPC sections.

A comparison of the results for pooling and mixture with the respective results for individual models (tables 7 and 8) shows that replacing data from the same task by data from related tasks decreases translation performance in almost all cases. The exception is the title model that benefits from pooling and mixing with both abstracts and claims due to their richer data structure.

### 4.3 Multi-task minimum error rate training

In contrast to task pooling and task mixtures, the specific setting addressed by multi-task minimum error rate training is one in which the generative SMT pipeline is not adaptable. Such situations arise if there are not enough data to train translation models or language models on the new tasks. However, we assume that there are enough parallel data available to perform meta-parameter tuning by minimum error rate training (MERT) (Och, 2003; Bertoldi et al., 2009) for each task.

A generic algorithm for multi-task learning can be motivated as follows: Multi-task learning aims to take advantage of commonalities shared among tasks by learning several independent but related tasks together. Information is shared between tasks through a joint representation and in-

| | tuning | | | | |
|---|---|---|---|---|---|
| test | individual | pooled | average | MMERT | MMERT-average |
| abstract | **0.3721** | 0.362 | $0.3657^{*+}$ | $\mathbf{0.3719}^{+}$ | $0.3685^{*+}$ |
| claim | 0.4711 | 0.4681 | $\mathbf{0.4749}^{*+}$ | $\mathbf{0.475}^{*+}$ | $0.4734^{*+}$ |
| title | 0.3228 | 0.3152 | $\mathbf{0.3326}^{*+}$ | $0.3268^{*+}$ | $\mathbf{0.3325}^{*+}$ |

Table 12: Multi-task tuning on text sections.

| | tuning | | | | |
|---|---|---|---|---|---|
| test | individual | pooled | average | MMERT | MMERT-average |
| A | 0.5187 | 0.5199 | $\mathbf{0.5213}^{*+}$ | 0.5195 | 0.5196 |
| B | 0.4877 | 0.4885 | $0.4908^{*+}$ | $0.4911^{*+}$ | $\mathbf{0.4921}^{*+}$ |
| C | **0.5214** | 0.5175 | $0.5199^{*+}$ | $\mathbf{0.5218}^{+}$ | $0.5162^{*+}$ |
| D | **0.4724** | **0.4730** | **0.4733** | **0.4736** | **0.4734** |
| E | 0.4666 | 0.4661 | $\mathbf{0.4679}^{*+}$ | $0.4669^{+}$ | $\mathbf{0.4685}^{*+}$ |
| F | 0.4794 | 0.4801 | $0.4811^{*}$ | $\mathbf{0.4821}^{*+}$ | $\mathbf{0.4830}^{*+}$ |
| G | 0.4596 | 0.4576 | $\mathbf{0.4607}^{+}$ | $\mathbf{0.4606}^{+}$ | $\mathbf{0.4610}^{*+}$ |
| H | 0.4573 | 0.4560 | **0.4578** | $\mathbf{0.4581}^{+}$ | $\mathbf{0.4581}^{+}$ |

Table 13: Multi-task tuning on IPC sections.

troduces an inductive bias. Evgeniou and Pontil (2004) propose a regularization method that balances task-specific parameter vectors and their distance to the average. The learning objective is to minimize task-specific loss functions $l_d$ across all tasks $d$ with weight vectors $w_d$, while keeping each parameter vector close to the average $\frac{1}{D}\sum_{d=1}^{D} w_d = w_{\text{avg}}$. This is enforced by minimizing the norm (here the $\ell_1$-norm) of the difference of each task-specific weight vector to the avarage weight vector.

$$\min_{w_1,\ldots,w_D} \sum_{d=1}^{D} l_d(w_d) + \lambda \sum_{d=1}^{D} \|w_d - w_{\text{avg}}\|_1 \quad (1)$$

The MMERT algorithm is given in figure 1. The algorithm starts with initial weights $w^{(0)}$. At each iteration step, the average of the parameter vectors from the previous iteration is computed. For each task $d \in D$, one iteration of standard MERT is called, continuing from weight vector $w_d^{(t-1)}$ and minimizing translation loss function $l_d$ on the data from task $d$. The individually tuned weight vectors returned by MERT are then moved towards the previously calculated average by adding or subtracting a penalty term $\lambda$ for each weight component $w_d^{(t)}[k]$. If a weight moves beyond the average, it is clipped to the average value. The process is iterated until a stopping criterion is met, e.g. a threshold on the maximum change in the average weight vector. The parameter $\lambda$ controls the influence of the regularization. A larger $\lambda$ pulls the weights closer to the average, a smaller $\lambda$ leaves more freedom to the individual tasks.

MMERT($w^{(0)}, D, \{l_d\}_{d=1}^{D}$):
**for** $t = 1, \ldots, T$ **do**
  $w_{\text{avg}}^{(t)} = \frac{1}{D}\sum_{d=1}^{D} w_d^{(t-1)}$
  **for** $d = 1, \ldots, D$ **parallel do**
    $w_d^{(t)} = \text{MERT}(w_d^{(t-1)}, l_d)$
    **for** $k = 1, \ldots, K$ **do**
      **if** $w[k]_d^{(t)} - w_{\text{avg}}^{(t)}[k] > 0$ **then**
        $w_d^{(t)}[k] = \max(w_{\text{avg}}^{(t)}[k], w_d^{(t)}[k] - \lambda)$
      **else if** $w_d^{(t)}[k] - w_{\text{avg}}^{(t)}[k] < 0$ **then**
        $w_d^{(t)}[k] = \min(w_{\text{avg}}^{(t)}[k], w_d^{(t)}[k] + \lambda)$
      **end if**
    **end for**
  **end for**
**end for**
**return** $w_1^{(T)}, \ldots, w_D^{(T)}, w_{\text{avg}}^{(T)}$

Figure 1: Multi-task MERT.

The weight updates and the clipping strategy can be motivated in a framework of gradient descent optimization under $\ell_1$-regularization (Tsuruoka et al., 2009). Assuming MERT as algorithmic minimizer[11] of the loss function $l_d$ in equation 1, the weight update towards the average follows from the subgradient of the $\ell_1$ regularizer. Since $w_{\text{avg}}^{(t)}$ is taken as average over weights $w_d^{(t-1)}$ from the step before, the term $w_{\text{avg}}^{(t)}$ is constant with respect to $w_d^{(t)}$, leading to the following subgradient (where $\text{sgn}(x) = 1$ if $x > 0$, $\text{sgn}(x) = -1$ if $x < 0$, and $\text{sgn}(x) = 0$ if $x = 0$):

$$\frac{\partial}{\partial w_r^{(t)}[k]} \lambda \sum_{d=1}^{D} \left\| w_d^{(t)} - \frac{1}{D} \sum_{s=1}^{D} w_s^{(t-1)} \right\|_1$$
$$= \lambda \ \text{sgn}\left( w_r^{(t)}[k] - \frac{1}{D} \sum_{s=1}^{D} w_s^{(t-1)}[k] \right).$$

Gradient descent minimization tells us to move in the opposite direction of the subgradient, thus motivating the addition or subtraction of the regularization penalty. Clipping is motivated by the desire to avoid oscillating parameter weights and in order to to enforce parameter sharing.

Experimental results for multi-task MERT (**MMERT**) are reported for both dimensions of patent tasks. For the IPC sections we trained a pooled model on 1,000,000 sentences sampled from abstracts and claims from all sections. We did not balance the sections but kept their original distribution, reflecting a real-life task where the distribution of sections is unknown. We then extend this experiment to the structural dimension. Since we do not have an intuitive notion of a natural distribution for the text sections, we train a balanced pooled model on a corpus composed of 170,000 sentences each from abstracts, claims and titles, i.e. 510,000 sentences in total. For both dimensions, for each task, we sampled 2,000 parallel sentences for development, development-testing, and testing from patents that were published in different years than the training data.

We compare the multi-task experiments with two baselines. The first baseline is individual task learning, corresponding to standard separate MERT tuning on each section (**individual**). This results in three separately learned weight vectors for each task, where no information has been shared between the tasks. The second baseline simulates the setting where the sections are not differentiated at all. We tune the model on a pooled development set of 2,000 sentences that combines the same amount of data from all sections (**pooled**). This yields a single joint weight vector for all tasks optimized to perform well across all sections. Furthermore, we compare multi-task MERT tuning with two parameter averaging methods. The first method computes the arithmetic mean of the weight vectors returned by the individual baseline for each weight component, yielding a joint average vector for all tasks (**average**). The second method takes the last average vector computed during multi-task MERT tuning (**MMERT-average**).[12]

Tables 12 and 13 give the results for multi-task learning on text and IPC sections. The latter results have been presented earlier in Simianer et al. (2011). The former table extends the technique of multi-task MERT to the structural dimension of patent SMT tasks. In all experiments, the parameter $\lambda$ was adjusted to 0.001 after evaluating different settings on a development set. The best result on each section is indicated in bold face; * indicates significance with respect to the individual baseline, + the same for the pooled baseline. We observe statistically significant improvements of 0.5 to 1% BLEU over the individual baseline for claims and titles; for abstracts, the multi-task variant yields the same result as the baseline, while the averaging methods perform worse. Multi-task MERT yields the best result for claims; on titles, the simple average and the last MMERT average dominate. Pooled tuning always performs significantly worse than any other method, confirming that it is beneficial to differentiate between the text section sections.

Similarly for IPC sections, small but statistically significant improvements over the individual and pooled baselines are achieved by multi-task tuning and averaging over IPC sections, excepting C and D. However, an advantage of multi-task tuning over averaging is hard to establish.

Note that the averaging techniques implicitly benefit from a larger tuning set. In order to ascertain that the improvements by averaging are not

[11] MERT as presented in Och (2003) is not a gradient-based optimization techniquem, thus MMERT is strictly speaking only "inspired" by gradient descent optimization.

[12] The aspect of averaging found in all of our multi-task learning techniques effectively controls for optimizer instability as mentioned in Clark et al. (2011).

| test | pooled-6k | significance |
|---|---|---|
| abstract | 0.3628 | < |
| claim | 0.4696 | < |
| title | 0.3174 | < |

Table 14: Multi-task tuning on 6,000 sentences pooled from text sections. "<" denotes a statistically significant difference to the best result.

simply due to increasing the size of the tuning set, we ran a control experiment where we tuned the model on a pooled development set of $3 \times 2,000$ sentences for text sections and on a development set of $8 \times 2,000$ sentences for IPC sections. The results given in table 14 show that tuning on a pooled set of 6,000 text sections yields only minimal differences to tuning on 2,000 sentence pairs such that the BLEU scores for the new pooled models are still significantly lower than the best results in table 12 (indicated by "<"). However, increasing the tuning set to 16,000 sentence pairs for IPC sections makes the pooled baseline perform as well as the best results in table 13, except for two cases (indicated by "<") (see table 15). This is due to the smaller differences between best and worst results for tuning on IPC sections compared to tuning on text sections, indicating that IPC sections are less well suited for multi-task tuning than the textual domains.

| test | pooled-16k | significance |
|---|---|---|
| A | 0.5177 | < |
| B | 0.4920 | |
| C | 0.5133 | < |
| D | 0.4737 | |
| E | 0.4685 | |
| F | 0.4832 | |
| G | 0.4608 | |
| H | 0.4579 | |

Table 15: Multi-task tuning on 16,000 sentences pooled from IPC sections. "<" denotes a statistically significant difference to the best result.

## 5 Conclusion

The most straightforward approach to improve machine translation performance on patents is to enlarge the training set to include all available data. This question has been investigated by Tinsley et al. (2010) and Utiyama and Isahara (2007). A caveat in this situation is that data need to be from the general patent domain, as shown by the inferior performance of a large Europarl-trained model compared to a small patent-trained model.

The goal of this paper is to analyze patent data along the topical dimension of IPC classes and along the structural dimension of textual sections. Instead of trying to beat a pooling baseline that simply increases the data size, our research goal is to investigate whether different subtasks along these dimensions share commonalities that can fruitfully be exploited by multi-task learning in machine translation. We thus aim to investigate the benefits of multi-task learning in realistic situations where a simple enlargement of training data is not possible.

Starting from baseline models that are trained on individual tasks or on data pooled from all tasks, we apply mixtures of translation models and multi-task MERT tuning to multiple patent translation tasks. We find small, but statistically significant improvements for multi-task MERT tuning and parameter averaging techniques. Improvements are more pronounced for multi-task learning on textual domains than on IPC domains. This might indicate that the IPC sections are less well delimitated than the structural domains. Furthermore, this is owing to the limited expressiveness of a standard linear model including 14-20 features in tuning. The available features are very coarse and more likely to capture structural differences, such as sentence length, than the lexical differences that differentiate the semantic domains. We expect to see larger gains due to multi-task learning for discriminatively trained SMT models that involve very large numbers of features, especially when multi-task learning is done in a framework that combines parameter regularization with feature selection (Obozinski et al., 2010). In future work, we will explore a combination of large-scale discriminative training (Liang et al., 2006) with multi-task learning for SMT.

## Acknowledgments

This work was supported in part by DFG grant "Cross-language Learning-to-Rank for Patent Retrieval".

## References


Nicola Bertoldi and Marcello Federico. 2009. Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the 4th EACL Workshop on Statistical Machine Translation*, Athens, Greece.

Nicola Bertoldi, Barry Haddow, and Jean-Baptiste Fouet. 2009. Improved minimum error rate training in Moses. *The Prague Bulletin of Mathematical Linguistics*, 91:7–16.

Fabienne Braune and Alexander Fraser. 2010. Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*, Beijing, China.

Alexandru Ceauşu, John Tinsley, Jian Zhang, and Andy Way. 2011. Experiments on domain adaptation for patent machine translation in the PLuTO project. In *Proceedings of the 15th Conference of the European Assocation for Machine Translation (EAMT 2011)*, Leuven, Belgium.

Olivier Chapelle, Pannagadatta Shivaswamy, Srinivas Vadrevu, Kilian Weinberger, Ya Zhang, and Belle Tseng. 2011. Boosted multi-task learning. *Machine Learning*.

Jonathan Clark, Chris Dyer, Alon Lavie, and Noah Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL'11)*, Portland, OR.

Hal Daumé. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07)*, Prague, Czech Republic.

Mark Dredze, Alex Kulesza, and Koby Crammer. 2010. Multi-domain learning by confidence-weighted parameter combination. *Machine Learning*, 79:123–149.

Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Analysis of translation model adaptation in statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT'10)*, Paris, France.

Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi-task learning. In *Proceedings of the 10th ACM SIGKDD conference on knowledge discovery and data mining (KDD'04)*, Seattle, WA.

Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. 2008. IRSTLM: an open source toolkit for handling large scale language models. In *Proceedings of Interspeech*, Brisbane, Australia.

Jenny Rose Finkel and Christopher D. Manning. 2009. Hierarchical bayesian domain adaptation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT'09)*, Boulder, CO.

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic.

George Foster, Pierre Isabelle, and Roland Kuhn. 2010. Translating structured documents. In *Proceedings of the 9th Conference of the Association for Machine Translation in the Americas (AMTA 2010)*, Denver, CO.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation (WMT'11)*, Edinburgh, UK.

Daniel Kifer, Shain Ben-David, and Johannes Gehrke. 2004. Detecting change in data streams. In *Proceedings of the 30th international conference on Very large data bases*, Toronta, Ontario, Canada.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Birch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, Prague, Czech Republic.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X*, Phuket, Thailand.

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proceedings of the joint conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING-ACL'06)*, Sydney, Australia.

Guillaume Obozinski, Ben Taskar, and Michael I. Jordan. 2010. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20:231–252.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the Human Language Technology Conference and the 3rd Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'03)*, Edmonton, Cananda.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. Technical Report IBM Research Division Technical Report, RC22176 (W0190-022), Yorktown Heights, N.Y.

Ariadna Quattoni, Xavier Carreras, Michael Collins, and Trevor Darrell. 2009. An efficient projection for $\ell_{1,\infty}$ regularization. In *Proceedings of the 26th International Conference on Machine Learning (ICML'09)*, Montreal, Canada.

Stefan Riezler and John Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL-05 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, Ann Arbor, MI.

Holger Schwenk. 2008. Investigations on large-scale lightly-supervised training for statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT'08)*, Hawaii.

Patrick Simianer, Katharina Wäschle, and Stefan Riezler. 2011. Multi-task minimum error rate training for SMT. *The Prague Bulletin of Mathematical Linguistics*, 96:99–108.

Matthew Snover, Bonnie Dorr, and Richard Schwartz. 2008. Language and translation model adaptation using comparable corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'08)*, Honolulu, Hawaii.

John Tinsley, Andy Way, and Paraic Sheridan. 2010. PLuTO: MT for online patent translation. In *Proceedings of the 9th Conference of the Association for Machine Translation in the Americas (AMTA 2010)*, Denver, CO.

Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou. 2009. Stochastic gradient descent training for $\ell_1$-regularized log-linear models with cumulative penalty. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP'09)*, Singapore.

Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. 2007. Transductive learning for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL'07)*, Prague, Czech Republic.

Masao Utiyama and Hitoshi Isahara. 2007. A Japanese-English patent parallel corpus. In *Proceedings of MT Summit XI*, Copenhagen, Denmark.

Bing Zhao, Matthias Eck, and Stephan Vogel. 2004. Language model adaptation for statistical machine translation with structured query models. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING'04)*, Geneva, Switzerland.