sebis

TUM

Topic Classification for Clauses in Terms of Services with Machine Learning

Bachelor’s Thesis Final Presentation – Jan Robin Geibel – 05.10.2020

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

Outline

Motivation | Terms of Services

- A **growing number of Terms of Service agreements** are entered into as more and **more goods and services are being bought online**.
- However, studies indicate that most **consumers do not read Terms of Services** [e.g., 3, 4].

➡ The majority of consumers conduct numerous transactions on a daily basis that are governed by Terms of Services without knowing their contents.

Motivation | The Thesis

Topic Classification for Clauses in Terms of Services with Machine Learning

Exploration of different Machine Learning methods to **automatically identify the topic** being addressed by **individual clauses of Terms of Services**

Related Work

- **Machine Learning** has been repeatedly applied in the **context of legal contracts and proceedings**.

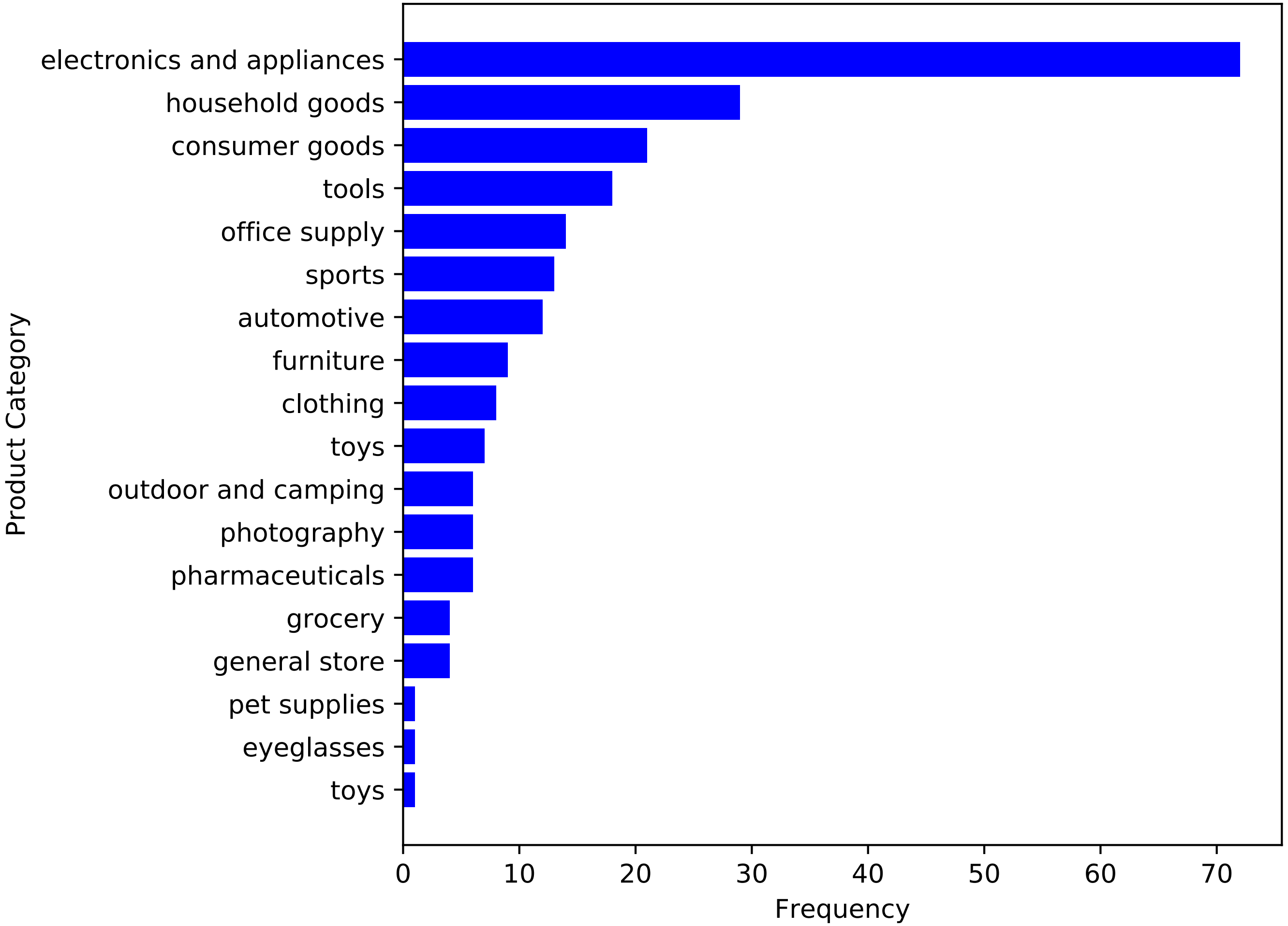
- Examples include
  - prediction of court decisions [e.g. 7]
  - extraction of arguments from legal documents [e.g. 8]
  - detection of claims in legal judgments [e.g. 9]
  - **classification the clauses of online consumer contracts according to their contents and their fairness** [e.g. 10,11,12]
  - **classification algorithms for clauses in privacy policies of online platforms** [e.g.13]

The Corpus | Origin of The Data

The Data

- **5020 clauses in German** from **142 e-commerce shops**.
- The **majority** of these are **located in Germany** with the exception of
  - **one** being headquartered in the **UK**,
  - **one** in the **Netherlands**,
  - **one** in **Luxemburg**
  - and **two** in the **Czech Republic**.

The Corpus | Origin of The Data



Nature of products sold by e-commerce shops the data was obtained from

General store: e-commerce platform offering a large variety of product categories

The Corpus | The Labeling Process

- **The clauses** and the **associated information** (incl. information about the e-commerce shops they were obtained from) were **manually collected** in an Excel file.

- **The classes** were **partly provided by the SEBIS Chair** and **partly derived in an iterative manner** by repeatedly grouping clauses according to their contents.

- The information collected for each clause:

Clause ID	File Number	Company	Paragraph Title	Paragraph Text	Clause Title	Clause Text	Clause Label 1	Clause Label 2

- The clauses were **labeled within the excel file** and later **exported to CSV** format.

The Corpus | The Classes

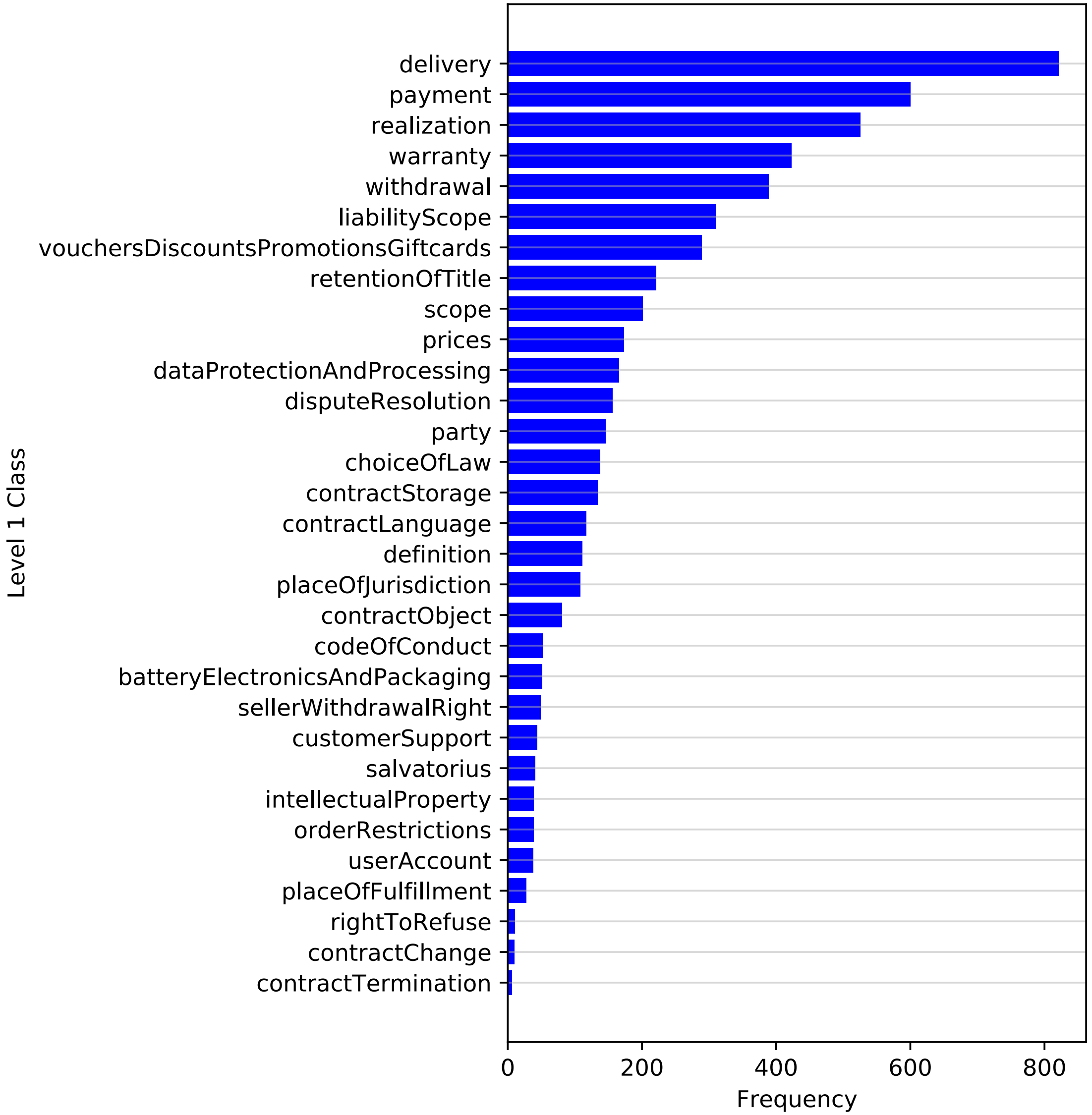
- The classification follows a **hierarchical approach**:
  - rather **broad labels (level 1 hereafter)** in a first step
  - which are in some cases then further subdivided into **more granular classes (level 2 hereafter)**

- Example:
  - **warranty**
    - **warranty:contractualClaims**
    - **warranty:exclusion**
    - **warranty:lapse**
    - **warranty:legalClaims**

- Clauses are **distinguishes based on their topics**

- **No assessment** whether their **legal implications** are the same is made

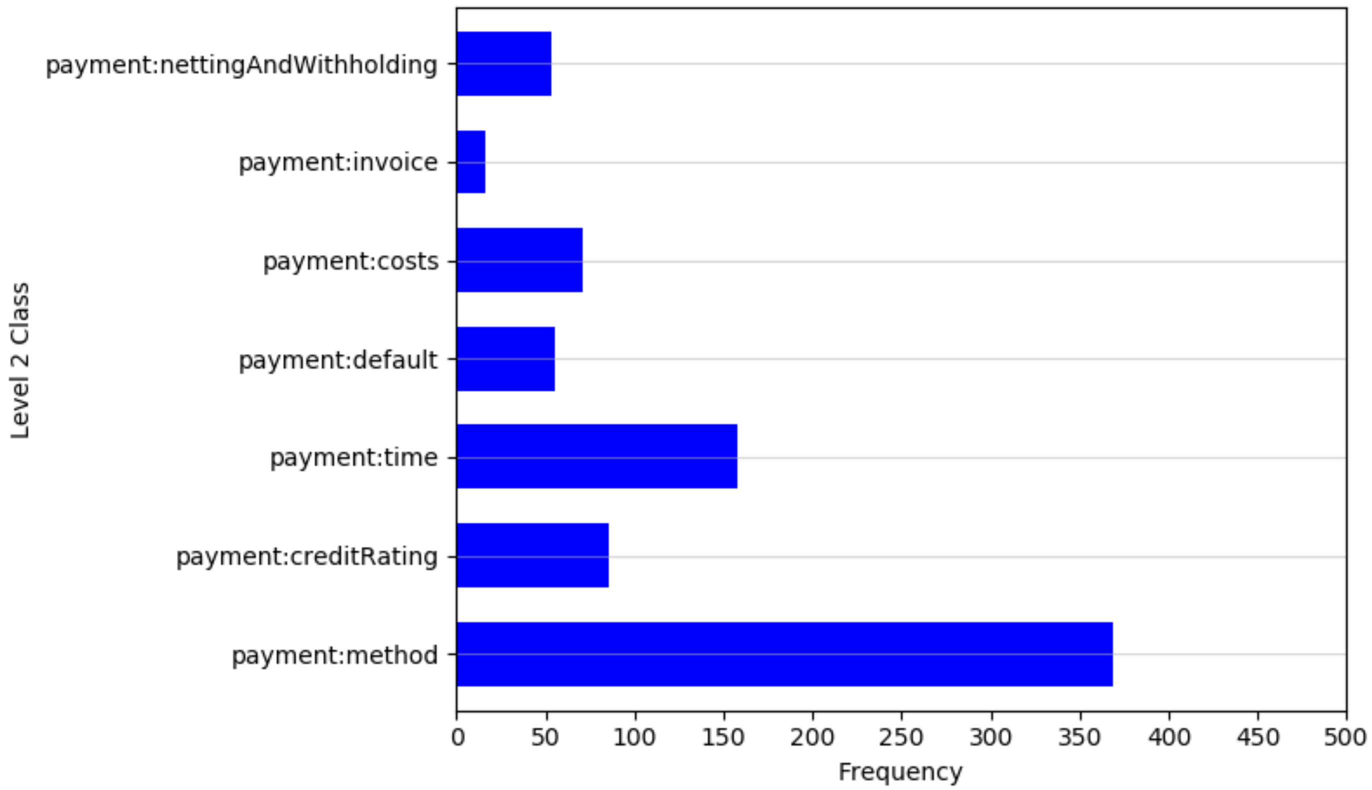
The Corpus | Distribution of Clauses Among Classes – Level 1



Distribution of clauses among level 1 classes

- Distribution (level 1) is **heavily skewed**
- *delivery*, *payment*, *realization*, *warranty* and *withdrawal* make up **about half of the labels given to clauses**

The Corpus | Distribution of Clauses Among Classes – Level 2



Distribution of Level 2 classes within Level 1 class payment

- **Distribution** (level 2) is similarly **concentrated**
- **45.6%** were the label ***payment:method***

Methodology | The Classification Problem

- **A clause** of a terms of service agreement **may address a variety of topics at once**

- The problem at hand is an instance of **multi-label classification**

- **Goal for each clause:**
  - arrive at a **ranking** or **probability estimate** for every individual label
  - **assign every label above certain threshold** (derived by evaluation on validation set)

Methodology | Data Preprocessing

1. The corpus is **tokenized** and **stopwords are removed**
   - **tokenization**: **breaking up** of text data into **individual components**, e.g. words
   - **stopwords**: words carrying **little information** [14]

2. The data is **lemmatized** and **special characters are removed**
   - **certain characters** such as § or € are **intentionally not removed**
   - **lemmatization:** different **varieties of a single word** which carry the **same semantic meaning** are **consolidated** [15]

Methodology | Feature Engineering

- **TF-IDF** (Term Frequency-Inverse Document Frequency):
  - **TF** (Term frequency): **number of times** a term appears in the corpus
  - **IDF** (inverse document frequency): **little significan**ce given to **words occurring very frequently** throughout the corpus – **great significance** given to those **occurring rarely** [14]

- **Word Embeddings**:
  - each **word** is **represented by a numerical vector**
  - **semantic relationship** between words = **geometric proximity** of their vectors [14]

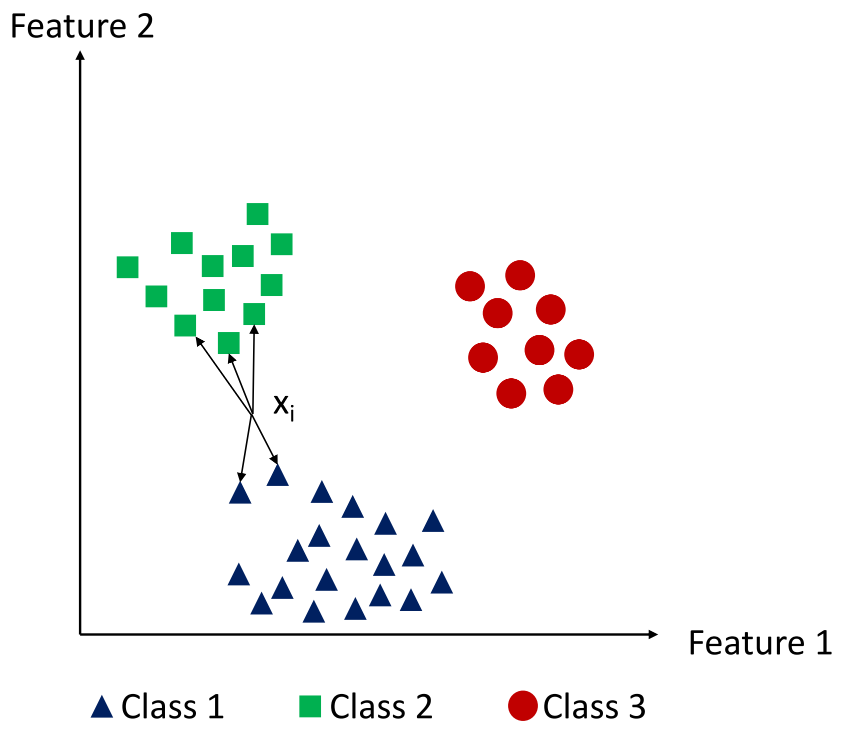
Methodology | Models

- **Support Vector Machine** (SVM):
  - **separate** the data points **by a hyperplane**
  - **maximize the distance** between **hyperplane** and **the data points** of the two classes on either side of it [16]

- **Logistic Regression** (LR):
  - **assign probabilities** to a data point being in either of the classes
  - **not directly predicting** the class it belongs to [15]

Methodology | Models

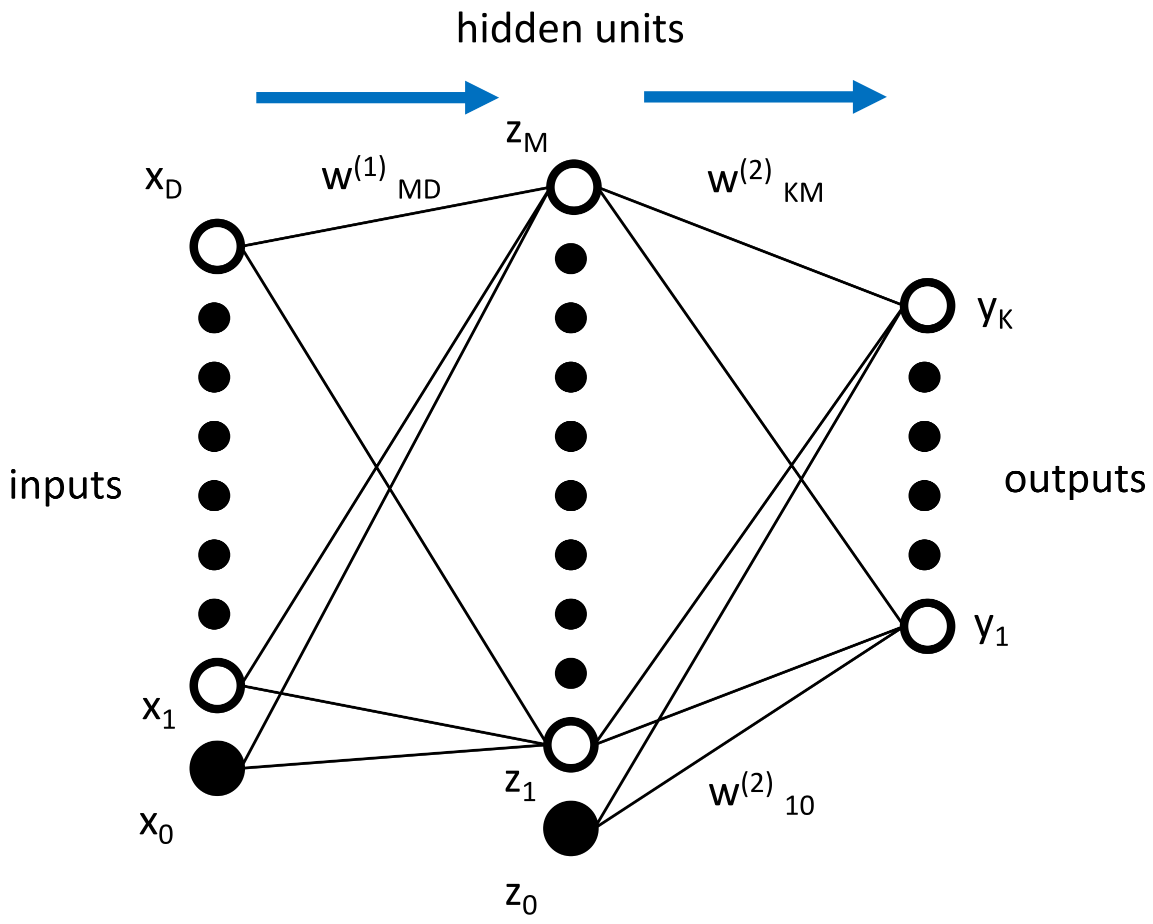
- **k-Nearest Neighbors** (kNN):
  - assign label **according to the k data points** of the training set
  - which are **most similar according to a predefined metric** [14]



k-Nearest Neighbors (based on [14])

Methodology | Models

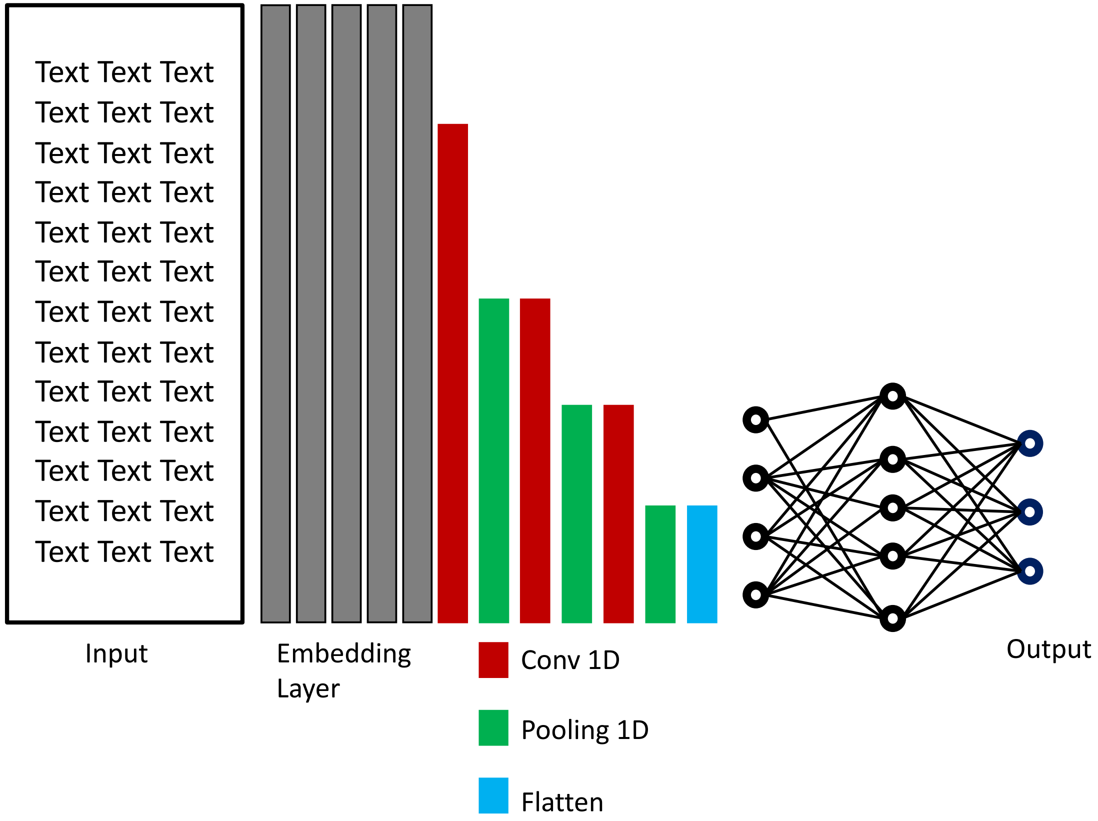
- **Multilayered Perceptron** (MLP):
  - consists of **input, output and hidden layers**
  - a succession of **linear combinations** and **non-linear functions** are used to **derive output values** [16]



Multilayered Perceptron (based on [17])

Methodology | Models

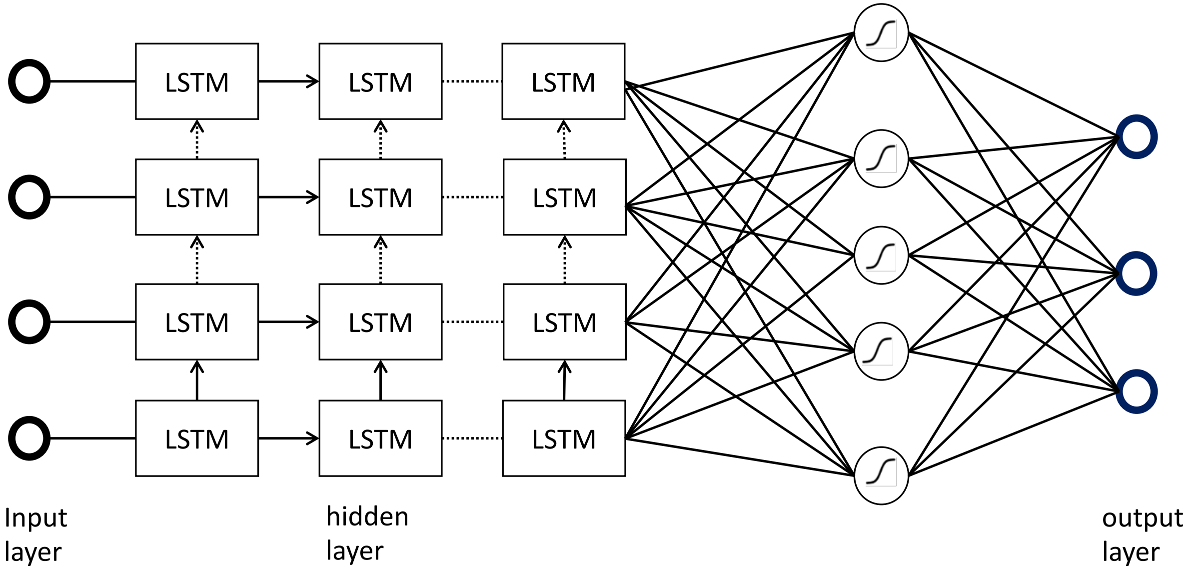
- **Convolutional Neural Networks** (CNN):
  - **deep learning algorithms** similar to MLP
  - originally **designed for Computer Vision**
  - intent to **leverage** that **closer pixels** of an image tent to be more **strongly correlated**
  - **subsets** of the data are **processed individually** [17]



Convolutional Neural Networks (based on [14])

Methodology | Models

- **Long Short-Term Memory** (LSTM):
  - neural networks used to **process sequential data such as text**
  - **consider early data points of a series** more than other architectures do [14]



Long Short-Term Memory (based on [14])

Methodology | Experimental Procedure

1. Establish baseline: **Train 4 classifiers** (SVC, LR, MLkNN and MLP) on **3011 clauses** (corpus version 1) to predict **level 1 labels**
   - using only **clause title and text** as input
   - using **paragraph title and text** as additional input
2. Train **same 4 classifiers** and a **CNN**, a **CNN with an embedding layer** and a **LSTM** on **5020 clauses** (corpus version 2) using **paragraph and clause information** as input to **predict level 1 labels**
3. Train **same 7 classifiers** plus
   - a **multi-input SVC** (TF-IDF, pre-trained SVC estimate for level 1 label, clause length)
   - a **multi-input MLP** (TF-IDF, pre-trained SVC estimate for level 1 label, clause length)
   - a **multi-input CNN** (TF-IDF, pre-trained SVC estimate for level 1 label)

Methodology | Experimental Procedure

Clause length:

- can indicate **correct level 2 label**
- can indicate when clause belongs to **multiple classes**

Pre-trained SVC estimate for level 1:

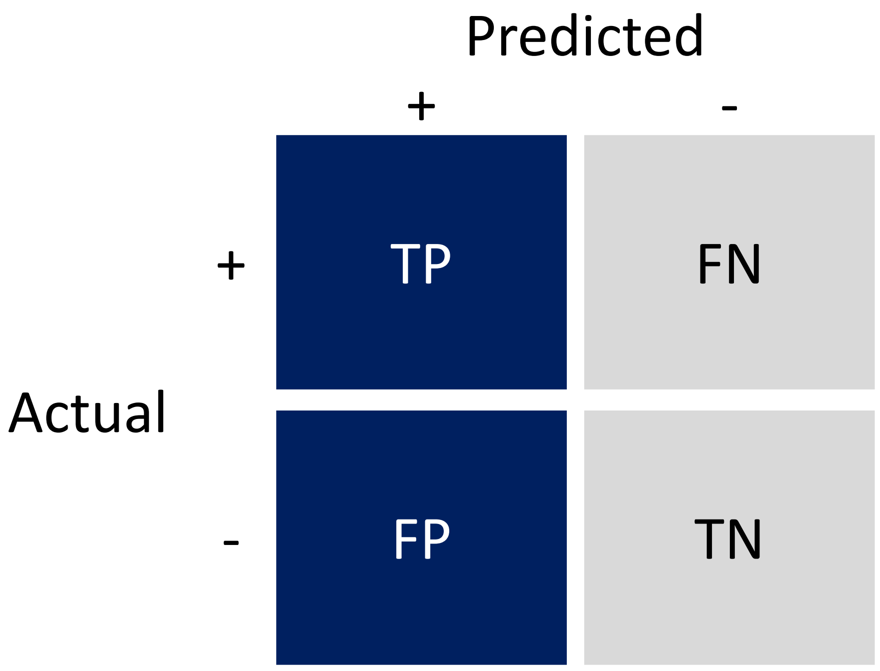
- can provide helpful information if **clause is routinely placed outside correct level 1 class**, e.g. *delivery:costs* is mistaken for *payment:costs*

Results | Evaluation Metrics

Micro-averaging was used throughout the project:

- results in an **average per data point** by considering the **decisions made for the clause over all classes** [14]

- TP: True Positives
- FP: False Positives
- FN: False Negatives
- TN: True Negatives



Confusion Matrix (based on [18])

Results | Evaluation Metrics

Accuracy:

- share of correctly classified data points in the entire data set

$$accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)}$$

Precision:

- share of data points that were correctly assigned to a class of all data points assigned to the class

$$precision = \frac{TP}{(TP + FP)}$$

Recall:

- share of data points that were correctly assigned to a class of all data points in the given class

$$recall = \frac{TP}{(TP + FN)}$$

F_1-Score:

- combination recall and precision

$$F_1 - score = \frac{2TP}{(2TP + FP + FN)}$$

1 **Results** | Train Classifiers on 3011 Clauses to Predict Level 1 Labels

Classifier	F_1-Score	Accuracy	Precision	Recall
SVC	0.879	0.794	0.905	0.853
Logistic Regression	0.729	0.534	0.649	0.832
MLkNN	0.821	0.75	0.858	0.787
MLP	0.872	0.794	0.922	0.826

Classifiers trained on version 1 corpus using clause information as input to predict level 1 labels - results on test set

Classifier	F_1-Score	Accuracy	Precision	Recall
SVC	0.903	0.837	0.908	0.897
Logistic Regression	0.763	0.572	0.687	0.858
MLkNN	0.852	0.779	0.926	0.79
MLP	0.889	0.826	0.932	0.85

Classifiers trained on version 1 corpus using clause and paragraph information as input to predict level 1 labels - results on test set

- Key observation: also providing paragraph information leads to significant **performance improvement**
- Possible explanation: proportion of clauses may **implicitly refer to the ones coming before** it - providing the **required context** in form of the **paragraph’s information**

➡ **Paragraph and clause information used as input for remainder of project**

2 **Results** | Train Classifiers on 5020 Clauses to Predict Level 1 Labels

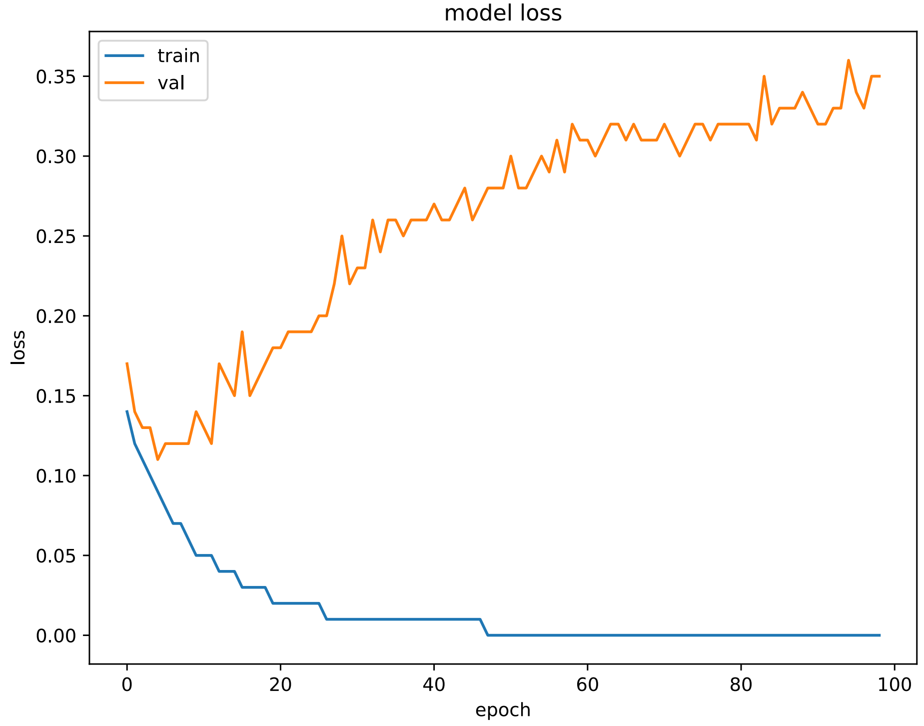
Classifier	F_1-Score	Accuracy	Precision	Recall
SVC	0.904	0.839	0.905	0.853
Logistic Regression	0.815	0.655	0.649	0.832
MLkNN	0.844	0.768	0.858	0.787
MLP	0.895	0.82	0.91	0.881
CNN	0.867	0.773	0.908	0.83
CNN Embedding Layer	0.568	0.468	0.583	0.553
LSTM	0.861	0.791	0.882	0.84

Classifiers trained on version 2 corpus using clause and as input to predict level 1 labels - results on test set

- Key observation: additional data only leads to **marginal improvement**
- Possible explanation:
  - clauses within a class is rather **homogeneous regarding their wording**
  - majority paragraphs/clauses likely **explicitly mention certain words indicating their topic**

2 **Results** | Train Classifiers on 5020 Clauses to Predict Level 1 Labels

- Key observation: CNN containing an embedding layer **performs poorly** despite being able to accurately classify the greater share of the **training data (0.993 F_1-Score)**
- Possible explanation: **overfitting to training data** (also indicated by learning curve)



CNN embedding layer trained on corpus version 2 to predict level 1 labels - loss during training process

3 **Results** | Train Classifiers on 5020 Clauses to Predict Level 2 Labels

Classifier	F_1-Score	Accuracy	Precision	Recall
SVC	0.834	0.706	0.805	0.866
Multi-input SVC	0.842	0.727	0.865	0.82
Logistic Regression	0.783	0.601	0.769	0.798
MLkNN	0.775	0.652	0.83	0.727
MLP	0.827	0.704	0.861	0.794
Multi-input MLP	0.837	0.708	0.863	0.812
CNN	0.791	0.643	0.854	0.736
CNN Embedding Layer	0.47	0.352	0.635	0.373
Multi-input CNN	0.82	0.695	0.878	0.769
LSTM	0.768	0.642	0.86	0.694

Classifiers trained on version 2 corpus using clause and paragraph information as input to predict level 2 labels - results on test set

3 **Results** | Train Classifiers on 5020 Clauses to Predict Level 2 Labels

Key observation:

- Providing **multiple inputs** to the SVC and MLP did **improve their results**
  - **not clear** which feature is responsible and **why** however
  - comparing the results per class also does **not allow for a clear conclusion**

- Large **discrepancy between precision and recall** in the results of the **LSTM and CNN**
  - e.g. the LSTM's **precision is 16.6 percentage points higher** than its recall
  - difference is even more apparent in the per class results:
    - **precision** for several classes was **1.0 (no false positives)**
    - **recall** was between **37.5 and 72.7 percentage points lower**

- CNN containing an embedding layer seems to also overfit for level 2 predictions

Conclusion | Key Observations

- **SVC and MLP perform remarkably** well in predicting level 1 and level 2 labels
- Providing **comparably little data is** sufficient to receive meaningful results
- Providing a **clause‘s length and an estimate of its level 1 label** can **improve performance**
- The **LSTM and CNN**
  - perform **well for level 1 predictions**
  - but their performance for **level 2 predictions** is significantly **less balanced**

Conclusion | Possible Next Steps

Improve the models' performance:

- use **pretrained word embedding** (watch out for overfitting)
- **optimize** deep learning **models' architecture** (sensitive to choice of batch size and hidden layers)
- **address** the potentially negative effects of the **severely unbalanced corpus**
- investigate further **approaches to hierarchical text classification**

Adapt corpus:

- Make even **more granular distinction** between clauses
- include clauses in **languages other than German**

Use results of classification in **more advanced application**. E.g. to make further qualitative assessment beyond a clause's topic.

B.Sc. Computer Science

Jan Robin Geibel

Technische Universität München
Faculty of Informatics
Chair of Software Engineering for Business
Information Systems

Boltzmannstraße 3
85748 Garching bei München

Tel +49.89.289.
Fax +49.89.289.17136

wwwmatthes.in.tum.de

Appendix | References

[1] „What Is a Terms of Service Agreement?“, upcounsel.com, accessed on 12.06.2020

[2] „7 von 10 Internetnutzern in der EU kaufen online“, Statistisches Bundesamt, accessed on 12.06.2020

[3] Obar, J. A. and Oueldorf-Hirsch, A., 2020. The biggest lie on the internet: Ignoring the privacy policies and terms of service policies of social networking services. *Information, Communication & Society*, 23 (1)

[4] Elshout, M., Elsen, M., Leenheer, J., Loos, M. and Luzak, J., 2016. Study on Consumers' Attitudes Towards terms Condiitions (T&Cs) Final Report. *Report for the European Commission, Consumers, Health, Agriculture and Food Executive Agency (Chafea) on behalf of Directorate-General for Justice and Consumers.*

[5] „AGB-Check – AI-Supported Legal Review of terms and Conditions to Strengthen Consumer Protection”, matthes.in.tum.de, accessed on 14.06.2020

[6] „Software Aided Analysis of Terms of Services”, matthes.in.tum.de, accessed on 14.06.2020

Appendix | References

[7] N. Aletras, D. Tsarapatsanis, D. Preotiuc-Pietro, and V. Lampos. Predicting judicial decisions of the european court of human rights: A natural language processing perspective. *PeerJ Computer Science*, 2:e93, 2016.

[8] M.-F. Moens, E. Boiy, R. M. Palau, and C. Reed. Automatic detection of arguments in legal texts. In *Proceedings of the 11th international conference on Artificial intelligence and law*, pages 225–230, 2007.

[9] M. Lippi, F. Lagioia, G. Contissa, G. Sartor, and P. Torroni. Claim detection in judgments of the eu court of justice. In *AI Approaches to the Complexity of Legal Systems*, pages 513–527. Springer, 2015.

[10] M. Lippi, P. Palka, G. Contissa, F. Lagioia, H.-W. Micklitz, Y. Panagis, G. Sartor, and P. Torroni. Automated detection of unfair clauses in online consumer contracts. In *JURIX*, pages 145–154, 2017.

[11] M. Lippi, P. Pałka, G. Contissa, F. Lagioia, H.-W. Micklitz, G. Sartor, and P. Torroni. Claudette: an automated detector of potentially unfair clauses in online terms of ser- vice. *Artificial Intelligence and Law*, 27(2):117–139, 2019.

Appendix | References

[12] F. Lagioia, F. Ruggeri, K. Drazewski, M. Lippi, H.-W. Micklitz, P. Torroni, and G. Sar- tor. Deep learning for detecting and explaining unfairness in consumer contracts. In *JURIX*, pages 43–52, 2019.

[13] G. Contissa, K. Docter, F. Lagioia, M. Lippi, H.-W. Micklitz, P. Pałka, G. Sartor, and P. Torroni. Claudette meets gdpr: Automating the evaluation of privacy policies using artificial intelligence. *Available at SSRN 3208596*, 2018.

[14] K.Kowsari, K.J.Meimandi, M.Heidarysafa, S.Mendu, L.Barnes, and D.Brown. Text classification algorithms: A survey. *Information*, 10:150, 2019.

[15] J.Plisson, N.Lavrac, and D. Mladenic. A rule based approach to word lemmatization. In *Proceedings of IS*, volume 3, pages 83–86, 2004.

[16] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. *Informatica*, 31(4):249–268, 2007.

[17] C. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.

Appendix | References

[18] J. Lever, M. Krzywinski, and N. Altman. Classification evaluation. *Nature Methods*, 13:603–604, 2016.