Entropy in Legal Language

ROLAND FRIEDRICH**, ETH Zurich, University of Zurich, Switzerland
MAURO LUZZATTO, ETH Zurich, Switzerland
ELLIOTT ASH, ETH Zurich, Switzerland

We introduce a novel method to measure word ambiguity, i.e. local entropy, based on a neural language model. We use the measure to investigate entropy in the written text of opinions published by the U.S. Supreme Court (SCOTUS) and the German Bundesgerichtshof (BGH), representative courts of the common-law and civil-law court systems respectively. We compare the local (word) entropy measure with a global (document) entropy measure constructed with a compression algorithm. Our method uses an auxiliary corpus of parallel English and German to adjust for persistent differences in entropy due to the languages. Our results suggest that the BGH's texts are of lower entropy than the SCOTUS's. Investigation of low- and high-entropy features suggests that the entropy differential is driven by more frequent use of technical language in the German court.

CCS Concepts: • **Computing methodologies** → **Lexical semantics**; • **Mathematics of computing** → **Markov processes**; **Coding theory**.

Additional Key Words and Phrases: neural language models, NLP, word2vec, entropy, civil law, common law, judiciary, comparative law

ACM Reference Format:
Roland Friedrich, Mauro Luzzatto, and Elliott Ash. 2020. Entropy in Legal Language. In *Proceedings of the 2020 Natural Legal Language Processing (NLLP) Workshop, 24 August 2020, San Diego, US.* ACM, New York, NY, USA, 11 pages.

1 INTRODUCTION

The world's legal systems feature two major traditions which have spread to almost all countries. These systems are the "`civil law`" as the continuation and refinement of the Roman "`jus civile`", and the "`common law`", as it originated in England after the Norman conquest in 1066 [4]. To oversimplify somewhat, a broad distinction of the systems is that at civil law judges make decisions from codified rules, while in the common law judges make decisions based on previous decisions.

In civil law commentaries, cf. e.g. [21], it is argued that common law lacks a strong principled foundation. On this view, common law is not systematised and without a general "`strategy`" but is rather driven by "`trial and error`" on a case by case basis. On the other hand, common law permits (judges) to adapt novel, pioneering and innovative ideas or doctrines more easily, and, as Posner [22] argued, it could be economically more efficient. Some evidence suggests that nations that followed the common law system have had better growth prospects than civil-law countries [14], although whether this effect is causal is not well-established.

A proffered reason for the relative inefficiency of civil-law institutions is that it is too rigid and cannot adapt well to changing circumstances. Code-based decision-making requires complex legislation that is costly to maintain, decipher, apply, and revise. These points are anecdotal, and there is not much good empirical evidence about them. Addressing these issues empirically is difficult because you do not have both common-law and civil-law systems operating in the same

*

NLLP @ KDD 2020, August 24th, San Diego, US

country. They also tend to be in different languages; common-law countries tend to be English-speaking, while Latin-Language and German-Speaking countries tend to have civil law. Perhaps foremost, we lack good measures of the complexity of the law.

Our goal is to produce some new measures of legal complexity in a comparative framework. We draw on recent technologies in neural language modeling to produce a new measure of local entropy at the word level. We then map entropy levels across case texts in an English-speaking common law court (the U.S. Supreme Court) and a German-speaking civil law court (the German Bundesgerichtshof).

The U.S. Supreme Court (SCOTUS) and German Bundesgerichtshof (BGH) are the highest courts in the respective legal systems. They are also two of the most influential judiciaries in the broader system of international law. Within the common-law and civil-law traditions, the SCOTUS and BGH are perhaps the most influential high courts of the last century.

We investigate the legal writing style of both the U.S. Supreme Court (SCOTUS) and the Bundesgerichtshof (BGH) from an information theoretic perspective, based on a neural language model. Concretely, we build our method on top of Mikolov's et al. [18] word2vec, in order to measure empirically the entropy at the token level, i.e. the micro scale.

We ask whether the two legal systems which these courts represent can be discriminated, solely based on information theoretic measures. We find that the BGH tends to have lower entropy than the SCOTUS, reflecting greater use of low-entropy technical language. Finally, in the case of the U.S. Supreme Court we further investigate the temporal evolution of the entropy both at the micro and macro level, by recording universal compression rates.

2 RELATED WORK

2.1 Entropy in Language

Shannon [26] in his seminal paper "`Prediction and Entropy of Printed English`" initiated the information theoretic study of natural languages. Similar to a theoretical physics approach, Shannon applied the mathematical tools he had previously conceived to understand information. That paper has led to a rich literature on measuring the information content in written and spoken text.

In this literature, a common and useful assumption is that language is *regular* in the sense that the underlying stochastic data generating process is both stationary and ergodic, cf. e.g. [9]. Kontoyiannis et al. [13] discuss various estimators for the Shannon entropy rate of a stationary ergodic process, and apply them to English texts. Most notable is the Lempel–Ziv [27] algorithm, which consistently estimates the entropy lower bound for stationary ergodic processes.

A recent application of the Lempel-Ziv compression algorithm to compare languages is Montemurro and Zanette [19]. They quantify the contribution of word ordering across different linguistic families to see if different languages had different entropy properties. They find that the Kullback-Leibler divergence (difference in entropy) between shuffled and unshuffled texts is a structural constant across all languages considered.

A complementary paper comparing languages at the word level is Bentz et al. [2]. They undertake a series of computer experiments to measure the word entropy across more than 1000 languages. They use unigram entropies which they estimate statistically. They find that word entropies follow a narrow unimodal distribution.

Degaetano-Ortlieb and Teich [5] is an application looking at changes in language entropy over time in a technical setting. They investigate the linguistic development of scientific English, by analysing the Royal Society Corpus (RSC) and the Corpus of Late Modern English (CLMET) computationally. They consider n-gram language models (for $n = 3$) and track the temporal changes

Table 1. Details of the corpora

Corpus	tokens	sentences
BGH Zivilsenat	30,166	410,612
BGH Strafsenat	11,313	110,645
U.S. Supreme Court	35,060	673,287
EuroParl German	73,439	1,967,341
EuroParl English	43,571	1,967,341

of the Kullback-Leibler divergence, as a measure of local ambiguity. Their main finding is that Scientific English, as it emerged over time, resulted in an increasingly optimised code for written communication by specialists.

2.2 Quantitative Analysis of Law

Our paper adds to the emerging literature in computational legal studies. Exemplary of this literature is Carlson, Livermore and Rockmore [3], who study the writing style of the U.S. Supreme Court. Katz et al. [6] apply machine learning, combined with classical statistical methods, as a novel approach to predict the behaviour of the U.S. Supreme Court in a generalised, out-of-sample context.

Klingenstein, Hitchcock, and DeDeo [11] take an information-theory approach to legal cases. They present a large-scale quantitative analysis of transcripts of London's Old Bailey. They use the Jensen-Shannon divergence to show that trials for violent and nonviolent offenses become increasingly distinct. This divergence reflects broader cultural shifts starting around 1800.

The use of neural text embeddings in law is illustrated by Ash and Chen [1]. That paper investigates the use of legal language and judicial reasoning in federal appellate courts, by using tools from natural language processing (NLP) and dense vector representations. They show that the resulting vector space geometry contains information to distinguish court, time, and legal topics.

3 DATA AND METHODS

The code used in this paper is available at https://github.com/MauroLuzzatto/legal-entropy.

3.1 Data

Our analysis is based on the U.S. Supreme Court decisions from the years 1924 to 2013, and the decisions of the German Bundesgerichtshof (BGH), covering the years 2014 until 2019. We separated the BGH data into rulings of the Zivil- and Strafsenat (civil and criminal chambers).

Additionally, as a baseline, we use Koehn's [12] EuroParl parallel corpus in German and English, consisting of the proceedings of the European Parliament from 1996 to 2006.

Some summary tabulations on the scope of the corpus are reported in Table 1.

3.2 Pre-Processing

For our analysis we use Python as well as spaCy [8] and NLTK [17] as our language processing tool.

We apply a standard preprocessing steps in order to train the word2vec model in gensim – for details cf. [23]. As an exception we did not lemmatise and stem the tokens, and we kept capitalisation. This makes English and German texts more comparable.

We also used the phraser function from gensim to treat idiomatic bigrams, such as "New York", and trigrams, such as "New York City", as single tokens.

Deserving special mention is the determination of sentence boundaries, a challenging task in legal writing [25]. We found this especially in the BGH civil case corpus, and less pronounced for the U.S. Supreme Court and the EuroParl data. A multitude of abbreviations, dates and most importantly statues involve a "`dot`", leading to a significant number of erroneous sentence tokens when the standard NLTK sentence tokenizer is naively applied. Therefore, before using nltk. sent_tokenize we removed all "`dots`" which do not indicate a sentence boundary, by compiling a look-up table in order to use it in conjunction with regular expression operations (RegEx).

3.3 Measuring Local Entropy using a Neural Language Model

To train word embeddings we use Gensim's [23] Word2Vec implementation. Word2Vec is a popular word embedding algorithm which uses a neural language model to predict local word co-occurrence. During the model training, a vector of predictive weights is learned for each word in the vocabulary. These weight vectors can be interpreted as the geometric location of the word in a semantic space, where words that are near each other in the space are semantically related.

There are two architectural versions of Word2Vec, CBOW and SkipGram. Simplified, in a CBOW model the neighboring context words are embedded to predict a left-out target word. In a Skip-Gram model, the target word is embedded to predict whether a paired word is sampled from the context or randomly sampled from outside the context.

Once trained, the Word2Vec model gives a predicted probability distribution across words given a context. Out of the box, Gensim offers for the CBOW model a command which yields the probability of a word to be a centre (target) word, depending on the context words to be specified. For the purposes of this project, we implemented the SkipGram version with hierarchical softmax. This model can be considered as the (neural) generalisation of the classical n-gram. This serves as our base in order to determine the local entropies.[1]

The window size is a hyperparameter. Larger windows capture more semantic relations whereas smaller windows tend to convey syntactic information [10]. Our experiments showed that SkipGram for a small context (window) size, e.g. $|c| = 2$, showed better results than the default window size ($|c| = 5$).[2]

For the discussion of the local entropy calculation and its implementation, cf. Appendix A.

For the Kolmogorov-Smirnov test we used SciPy.

3.4 Measuring Global Entropy using Lempel-Ziv Compression

The second entropy measure we compute uses the Lempel-Ziv algorithm for sequential data. First, we compress the raw text using the gzip compression module interface in Python, with the compression level set to its maximum value (= 9).

We define the compression ratio, r_i, of an individual text, txt_i, as $r_i := \frac{|txt_i|}{|\text{gzip}(txt_i)|}$, where $||$ denotes the size as measured in bits. The inverse ratio r^{-1} yields the fraction of the compressed file in comparison to the original file. Note that $r_i > 0$ for all documents i and equivalently for the entire corpus. When considering compression rates for individual texts and the entire corpus, one should keep in mind the sub-additivity of the Shannon entropy.

[1] For a detailed discussion of predicting a context word from a target word, see https://stackoverflow.com/questions/45102484/predict-middle-word-word2vec.

[2] A recent experimental study for SkipGram models by Lison and Kutuzov [16], found that for semantic similarity tasks right-side contexts are more important than left-side contexts, at least for English, and that the average model performance was not significantly influenced by the removal of stop words.

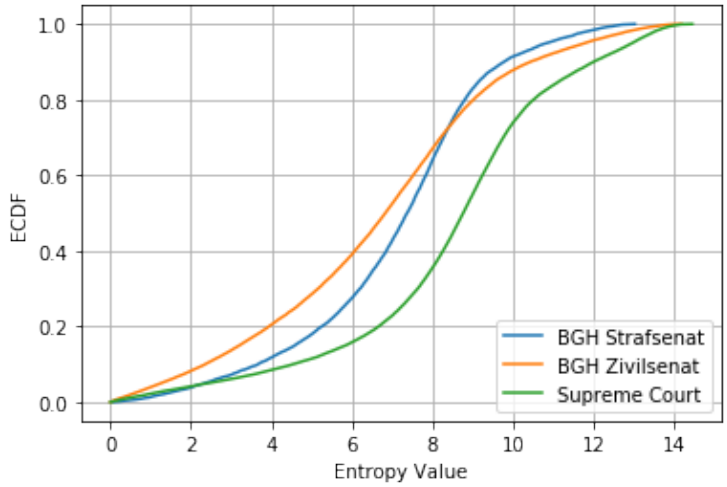

Fig. 1. Empirical cumulative distribution functions (ECDF) of the local entropy values for the BGH's Straf- and Zivilsenat and the U.S. Supreme Court, displaying the civil law-common law hysteresis.

4 RESULTS

4.1 Local Entropy of Words

Our first analysis is to compare the distributions of the word entropies across the different corpora. We would like to determine the differences in the distribution of the local entropy values of the language used by the BGH's Straf- and Zivilsenat and the U.S. Supreme Court. To this end, Figure 1 plots the respective empirical cumulative distribution functions $\text{ECDF}_{\text{BGH-Z}}$, $\text{ECDF}_{\text{BGH-Str}}$ and ECDF_{SC}.

As can be seen in the figure, in the interval $[0, 4]$ the distributions of the BGH's criminal chambers and the U.S. Supreme Court are similar, whereas for entropy values $t \geq 4$ we find that $\text{ECDF}_{\text{BGH-Str}}(t) > \text{ECDF}_{\text{SC}}(t)$, i.e. the Strafsenat's curve is strictly above the U.S. Supreme Court's.

Comparing the Zivilsenat to the U.S. Supreme Court we find that $\text{ECDF}_{\text{BGH-Z}}(t) - \text{ECDF}_{\text{SC}}(t) > 0$, for every $t \in [0, \max(\text{entropy(BGH-Z)})]$, i.e. the difference between the ECDF curves of the Zivilsenat and the U.S. Supreme Court is always strictly positive.

4.2 Adjusting for English-German Language Differences

We use the EuroParl German corpus and its aligned English translation as a baseline for two reasons. First, we want to gauge the quality of our local entropy method. Second, we would like to disentangle language-specific effects, i.e. English vs. German, when comparing the U.S. Supreme Court to the BGH.

Figure 2 demonstrates how the method behaves across languages using the parallel, sentence aligned EuroParl German and English corpora. As predicted by theory for a good translation, our method yields two highly identical probability distributions (Left Panel). As seen in the Right Panel, the empirical cumulative distribution functions of the local entropies are also very similar. It would be interesting to further study the influence of n-grams on the local entropy distribution of translations.

We quantified the distance between the empirical distribution functions of the EuroParl English and German corpora via the two-sided Kolmogorov–Smirnov test [7]. The null hypothesis H_0 states

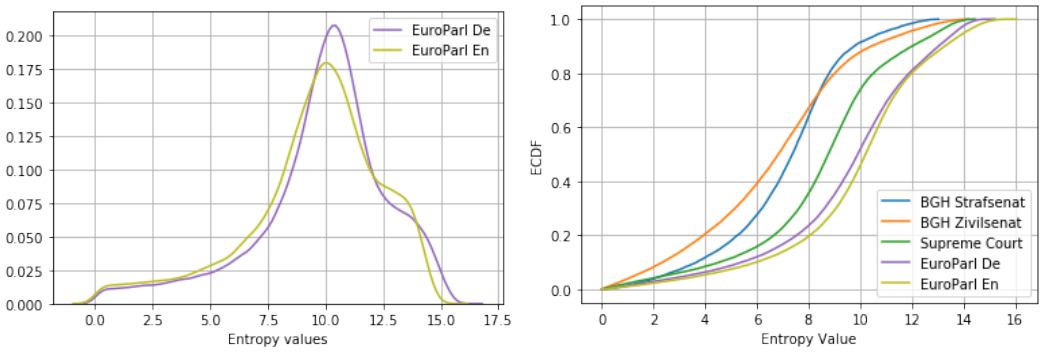

Fig. 2. Left Panel: Probability distributions of the local entropy values of the European Parliaments German proceedings (EuroParl de) and of its English translation (EuroParl en). Right Panel: Empirical cumulative distribution functions (ECDF) of the local entropy values for the BGH's Straf- and Zivilsenat, the U.S. Supreme Court, EuroParl Deutsch, and EuroParl English.

that two observed and stochastically independent samples are drawn from the same (continuous) distribution. We calculated the value of the ECDF in steps of $1/10$ in the interval $[0, 16]$, i.e. the range of the entropy values. The result for the D-statistics is 0.069 and for the two-tailed p-value 0.843, therefore we cannot reject H_0.

Second, the comparison with the baseline suggests, that as we hypothesised the use of German and English, respectively, in the courts has significantly less local entropy, as compared to the more colloquial and non technical (one might even argue scientific) use of the language in political speeches. This results in the strict local ambiguity order

$$\mathrm{ECDF}_{\text{BGH-Z}} \prec \mathrm{ECDF}_{\text{BGH-Str}} \prec \mathrm{ECDF}_{\text{SC}} \prec \mathrm{ECDF}_{\text{EP-de}} \sim \mathrm{ECDF}_{\text{EP-en}} .$$

4.3 Global Entropy of Documents

Now we produce the more global measure of entropy using the compression-based measure. We estimated the macroscopic entropy of the different corpora by compressing the entire raw text file for each and then calculating the corresponding inverse compression ratios, as described above. A higher value means that the corpus has higher entropy per segment of text. Put differently, a lower value means that there is relatively more structure or predictability in the underlying text features.

Table 2 reports the compression ratios for each corpus. As before, the values for the EuroParl corpora are almost identical, and they have the highest entropy rate. This likely reflects the broader diversity of issues covered in EuroParl relative to the law. The U.S. Supreme Corpus has a slightly lower entropy rate. Meanwhile, the BGH's Strafsenat and Zivilsenat corpora yield substantially lower values, with the BGH's civil courts having the lowest ratio of 0.283.

Next, we show how entropy varies over time in the SCOTUS data. Fig. 3 shows the inverse compression ratio entropy measures for the records of the U.S. Supreme Court in the last century. We can see that entropy has decreased since the 1950s, indicating an increase in the relative structure or predictability in the text.

This trend can be interpreted as a more formalised and standardised writing style. The shift could be due to the ongoing expansion of administrative (statutory) law in the U.S. system. Once statues are extensively used, the need for efficient methods of referral emerge, e.g. [§§ articles, sections, lit.,...], leading to a cryptic, pseudocode-like style of writing. This code-like, technical style was already extensively used by the BGH or the French Court of Cassation.

Corpus	Compression Ratio Entropy
EuroParl German	0.323
EuroParl English	0.322
U.S. Supreme Court	0.316
BGH Strafsenat	0.300
BGH Zivilsenat	0.283

Table 2. Inverse Compression Ratio Entropy, by Corpus. See Subsection 3.4 for method details.

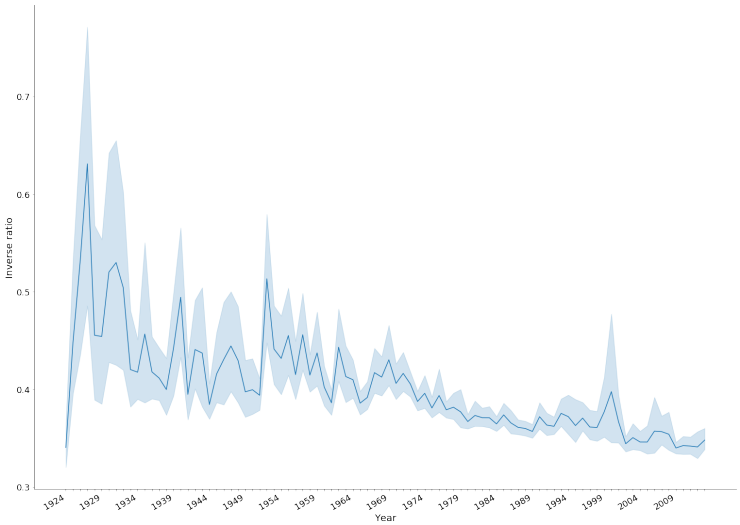

Fig. 3. Per document inverse gzip compression ratio of the U.S. Supreme Court for the period 1924 until 2013 (higher value means higher entropy).

4.4 Low-Entropy Words are Functional

To further substantiate the above ideas, we selected from each corpus (SCOTUS, BGH Zivil- and Strafsenat, EuroParl German and English) tokens with the lowest local entropy value ≤ 1. Fig. 4 includes word clouds for the lowest-entropy words in our vocabulary.

For the BGH (bottom left) one recognises key phrases from procedural law such as, e.g. 'zurückverweisen' (to send back a request). We see technical language for civil cases, such as 'Insolvenzverfahrens' (bankruptcy proceeding). For the SCOTUS, we see procedural, criminal and civil technical phrases such as 'beyond reasonable' and 'qualified immunity'. For the EuroParl data, the dominating lowest entropy phrases are procedural and related to the Parliament's sessions, such as, e.g. the German 'siehe_Protokoll' which corresponds to the English 'see_Minutes'.

The very low entropy words, serve as functional foundations in order to typify the respective environment and to set the tone. These reoccurring phrases have a very precise meaning, as the human reader recognises, and as quantitatively reflected in our neural model.

An in-depth analysis of the precise distribution of the local entropies along the different linguistic axes, and the broader syntactic and semantic categories, is left for a separate publication.

Fig. 4. Word clouds for Lowest-Entropy Words: Top left: EuroParl German. Top right: EuroParl English; Bottom left: BGH Zivilsenat. Bottom right: U.S. Supreme Court (SCOTUS).

5 CONCLUSION

Our analysis has shown that the writing style in civil law has lower relative entropy than the common law, at least in the important cases of the SCOTUS and BGH. We have shown this for two measures. First, local ambiguity, i.e. word entropy, produced using a neural language model, and second, global entropy produced from a compression ratio algorithm. Civil and common law writing styles are distinguishable on a purely information-theoretic base.

The results are helpful from the perspectives of history and social science. The original German legal doctrine is very much rooted in jurisprudence and has been strongly influenced, especially after the second half of the 19th century, by the development of natural sciences. This systematic approach is reflected in the writing style. Code-based legal writing requires, as argued above, efficient and standardised mechanisms of referencing, common to all scientific writing.

Our method innovates by using a neural language model, combined with data compression algorithms, in order to empirically determine both word and stylistic ambiguity, i.e. local and global entropy. This approach proves to be fruitful and could integrate naturally into future enhancements

of (deeper) neural language models. In future work these could provide an even finer spatio-temporally resolution of how information is distributed on different linguistic scales and time, ranging from the word to the corpus level.

In summary, our implementation and use of a local entropy measure, based on a neural language model, has led to striking results that contribute to an old debate on legal traditions. The contribution could be important both from a linguistic but also legal perspective. We foresee a broad range of further applications.

REFERENCES

[1] Elliott Ash and Daniel L. Chen. 2018. Mapping the Geometry of Law Using Document Embeddings.

[2] Christian Bentz, Dimitrios Alikaniotis, Michael Cysouw, and Ramon Ferrer-i Cancho. 2017. The Entropy of Words—Learnability and Expressivity across More than 1000 Languages. *Entropy* 19, 6 (Jun 2017), 275. https://doi.org/10.3390/e19060275

[3] Keith Carlson, Michael A. Livermore, and Daniel Rockmore. 2015-2016. A Quantitative Analysis of Writing Style on the U.S. Supreme Court. *Washington University Law Review* 93 (2015-2016), 1461.

[4] Joseph Dainow. 1966. The Civil Law and the Common Law: Some Points of Comparison. *The American Journal of Comparative Law* 15, 3 (1966), 419–435. http://www.jstor.org/stable/838275

[5] Stefania Degaetano-Ortlieb and Elke Teich. 2019. Toward an optimal code for communication: The case of scientific English. *Corpus Linguistics and Linguistic Theory* 0 (2019).

[6] MJ Bommarito DM Katz and J Blackman. 2017. A general approach for predicting the behavior of the Supreme Court of the United States. *PLoS ONE* 12, 4 (2017). https://doi.org/10.1371/journal.pone.0174698

[7] J. L. Hodges. 1958. The significance probability of the smirnov two-sample test. *Ark. Mat.* 3, 5 (01 1958), 469–486. https://doi.org/10.1007/BF02589501

[8] Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. (2017). To appear.

[9] D. Jurafsky and J. H. Martin. 2019. *Speech and Language Processing* (3 ed.). draft; https://web.stanford.edu/~jurafsky/slp3/.

[10] U. Kamath, J. Liu, and J. Whitaker. 2019. *Deep Learning for NLP and Speech Recognition*. Springer International Publishing. https://books.google.ch/books?id=8cmcDwAAQBAJ

[11] Sara Klingenstein, Tim Hitchcock, and Simon DeDeo. 2014. The civilizing process in London's Old Bailey. *Proceedings of the National Academy of Sciences* 111, 26 (2014), 9419–9424. https://doi.org/10.1073/pnas.1405984111 arXiv:https://www.pnas.org/content/111/26/9419.full.pdf

[12] Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*. AAMT, AAMT, Phuket, Thailand, 79–86. http://mt-archive.info/MTS-2005-Koehn.pdf

[13] I. Kontoyiannis, P. H. Algoet, Y. M. Suhov, and A. J. Wyner. 1998. Nonparametric entropy estimation for stationary processes and random fields, with applications to English text. *IEEE Transactions on Information Theory* 44, 3 (1998), 1319–1327.

[14] Rafael La Porta, Florencio Lopez-de Silanes, and Andrei Shleifer. 2008. The economic consequences of legal origins. *Journal of economic literature* 46, 2 (2008), 285–332.

[15] Omer Levy and Yoav Goldberg. 2014. Dependency-Based Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 302–308. https://doi.org/10.3115/v1/P14-2050

[16] Pierre Lison and Andrey Kutuzov. 2017. Redefining Context Windows for Word Embedding Models: An Experimental Study. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*. Association for Computational Linguistics, Gothenburg, Sweden, 284–288. https://www.aclweb.org/anthology/W17-0239

[17] Edward Loper and Steven Bird. 2002. NLTK: The Natural Language Toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

[18] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

[19] M.A. Montemurro and D. H. Zanette. 2011. Universal Entropy of Word Ordering Across Linguistic Families. *PLoS ONE* 6, 5 (2011).

[20] Frederic Morin and Yoshua Bengio. 2005. Hierarchical Probabilistic Neural Network Language Model. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, Robert G. Cowell and Zoubin Ghahramani (Eds.).

Society for Artificial Intelligence and Statistics, 246–252. http://www.iro.umontreal.ca/~lisa/pointeurs/hierarchical-nnlm-aistats05.pdf
[21] Marcel Alexander Niggli and Louis Frédéric Muskens. 2014. BSK StGB-Niggli/Muskens, Art. 11. In *Schweizerische Strafprozessordnung/Jugendstrafprozessordnung (StPO/JStPO)* (2 ed.), Marianne Heer Marcel Alexander Niggli and Hans Wiprächtiger (Eds.). Vol. 1. Helbing & Lichtenhahn, 3501.
[22] R.A. Posner. 2003. *Economic Analysis of Law.* Aspen Publishers. https://books.google.ch/books?id=gyUkAQAAIAAJ
[23] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.* ELRA, Valletta, Malta, 45–50. http://is.muni.cz/publication/884893/en.
[24] Xin Rong. 2014. word2vec Parameter Learning Explained. http://arxiv.org/abs/1411.2738 cite arxiv:1411.2738.
[25] George Sanchez. 2019. Sentence Boundary Detection in Legal Text. In *Proceedings of the Natural Legal Language Processing Workshop 2019.* Association for Computational Linguistics, Minneapolis, Minnesota, 31–38. https://doi.org/10.18653/v1/W19-2204
[26] C. E. Shannon. 1951. Prediction and Entropy of Printed English. *Bell System Technical Journal* 30, 1 (1951), 50–64. https://doi.org/10.1002/j.1538-7305.1951.tb01366.x arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.1538-7305.1951.tb01366.x
[27] J. Ziv and A. Lempel. 1977. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory* 23, 3 (1977), 337–343.

A THEORY

Here we give a theoretical description of the steps underlying our approach.

A.1 Preprocessing

Let C be a non-empty set, the corpus. For $n \in \mathbb{N}$, consider the map

$$\pi_n : C \to V_n$$

where V_n is the, possibly empty, set of n-grams (associated to C), which satisfy $V_k \cap V_l = \emptyset$, for $l \neq k$. Usually, the set of unigrams V_1, is called the vocabulary of the corpus C.

For a fixed $\nu \in \mathbb{N}$, set

$$\mathcal{V}_\nu := \bigcup_{n=1}^{\nu} V_n$$

which is the set of (two-sided) uni-, bi-, tri- up to ν-grams, and which, for ν large enough, yields an approximation (or pairwise disjoint decomposition) of the corpus C, which capture both syntactic and semantic information.[3] Then $\mathcal{V}_\nu$ is the (generalised) vocabulary up to order ν. The elements $w \in \mathcal{V}_\nu$, or $\mathcal{V}$ if ν is fixed and clear from the context, are **tokens** or n**-grams**, which might be considered as n-order words. We denote by $|\mathcal{V}|$ the size of $\mathcal{V}$, i.e. the number of pairwise different tokens.

The family of maps π_n, and hence the specific sets V_n, determine the preprocessing of the corpus data.

A.2 Local entropy from word2vec

The word2vec framework consists of a bundle of mathematical objects [18, 24]. First, it defines a dense Hilbert space representation,

$$\begin{aligned} \text{word2vec} : \mathcal{V} &\to \mathbb{R}^N, \\ w &\mapsto h_w, \end{aligned}$$

[3] More general, i.e. functional neighbourhoods are of course possible, e.g. based on grammatical information, as considered by Levy and Goldberg [15].

where $N \in \mathbb{N}$ is the dimension of the coordinate space, which is a hyper-parameter of the model. Let $\mathfrak{P}(\mathcal{V})$ be denote the set of discrete probability distributions on $\mathcal{V}$. Then, there exists a map

$$\begin{aligned} f_{w2v} : \mathcal{V} &\rightarrow \mathfrak{P}(\mathcal{V}), \\ w &\mapsto \mu_w, \end{aligned}$$

which associates to every token w a probability distribution μ_w, namely the posterior (multinomial) distribution. The **local entropy** or **ambiguity** is the map

$$\begin{aligned} H : \mathcal{V} &\rightarrow \mathbb{R}_+, \\ w &\mapsto H(\mu_w), \end{aligned}$$

which assigns to every token w the Shannon entropy of the corresponding probability distribution μ_w. The posterior distribution is given by a Boltzmann distribution (softmax).

It is calculated as follows. Let W be the $|\mathcal{V}| \times N$ input weight matrix from the input layer to the hidden layer and $\widetilde{W}$ the $N \times |\mathcal{V}|$ weight matrix from the hidden layer to the output layer in the SkipGram model with hierarchical softmax.

Every token $w_i \in \mathcal{V}$ determines a pair of vectors $(v_i, \tilde{v}_i)$, the input vector v_i and the output vector $\tilde{v}_i$, which are given by the ith row of W and the ith column of $\widetilde{W}$, respectively.

Let

$$Z_i := \sum_{j=1}^{|\mathcal{V}|} e^{\langle \tilde{v}_j | v_i \rangle} \tag{1}$$

be the local partition function corresponding to the target w_i, with the sum taken over all tokens $w_j \in \mathcal{V}$. (We use the bra-ket notation).

For the SkipGram model with context c, the probability $\mu_{w_i}(w_o)$ of a token w_o being an actual c-context output word of w_i, is given by

$$p(w_o|w_i) := \mu_{w_i}(w_o) := \frac{1}{Z_i} e^{\langle \tilde{v}_o | v_i \rangle}. \tag{2}$$

Therefore, the local entropy of the target w_i (with context c) is given by

$$H(w_i) := H(\mu_{w_i}) = -\sum_{j=1}^{|\mathcal{V}|} p(w_j|w_i) \cdot \log_2(p(w_j|w_i)). \tag{3}$$

A.3 Gensim implementation

We implemented our local entropy calculation for the SkipGram model in gensim, with the following parameters: context window= 2, $N = 300$ and 30 training epochs with hierarchical softmax [20].

The output weight matrix $\widetilde{W}$, and the input weight matrix W, are stored by gensim in the files syn1 (for hierarchical softmax) and syn0, respectively. Note, if negative sampling is used, then the output weights are stored in syn1neg.