Textual Characteristics for Language Engineering

Mathias Bank†, Robert Remus‡, Martin Schierle*

†Pattern Science AG, 63579 Freigericht, Germany
‡Natural Language Processing Group, University of Leipzig, Germany
∗ Mercedes-Benz RD North America, Palo Alto, USA
m.bank@cid.biz, rremus@informatik.uni-leipzig.de, martin.schierle@daimler.com

Abstract

Language statistics are widely used to characterize and better understand language. In parallel, the amount of text mining and information retrieval methods grew rapidly within the last decades, with many algorithms evaluated on standardized corpora, often drawn from newspapers. However, up to now there were almost no attempts to link the areas of natural language processing and language statistics in order to properly characterize those evaluation corpora, and to help others to pick the most appropriate algorithms for their particular corpus. We believe no results in the field of natural language processing should be published without quantitatively describing the used corpora. Only then the real value of proposed methods can be determined and the transferability to corpora originating from different genres or domains can be estimated. We lay ground for a language engineering process by gathering and defining a set of textual characteristics we consider valuable with respect to building natural language processing systems. We carry out a case study for the analysis of automotive repair orders and explicitly call upon the scientific community to provide feedback and help to establish a good practice of corpus-aware evaluations.

Keywords: Textual Characteristics, Language Engineering, Language Statistics

1. Motivation

Language statistics and quantitative linguistics are widely used to study, characterize and better understand language, to help foreign learners or even to identify authors (Holmes, 1994). Těšitelová (1992) provides a comprehensive overview of the large pool of methods available today. Implicitly connected, natural language processing (NLP) methods often rely on statistical methods and machine learning algorithms, which in turn massively rely on certain *textual characteristics*, e.g. token frequencies, token distributions and token probability transitions. Still, textual characteristics of corpora used for training and testing such methods and algorithms are rarely analyzed and documented. We strongly believe the successful creation of real world NLP systems, i.e. the selection of appropriate methods and algorithms, is only possible if the respective text types are soundly understood. Furthermore, we believe scientific publications in NLP must clearly document language statistics of the used corpora. This is necessary because not all algorithms work equally on every text type and their portability may be questionable (Sekine, 1997; Escudero et al., 2000; Wang and Liu, 2011). Only by knowing the textual characteristics of a certain text type it is possible to estimate the transferability of proposed methods and hence assess their real value. To our best knowledge there is no previous work that uses language statistics to give guidance in building NLP systems, although this is a crucial part of every *language engineering* (Cunningham, 1999) process.
In the next Section, we select and present suitable language statistics. In Section 3. we apply them to English-language corpora from three different *genres*: news articles, web fora posts and automotive repair orders. In Section 4. we carry out a case study and demonstrate how textual characteristics may give guidance to select appropriate algorithms for a successful genre-specific information extraction system. Finally, we draw conclusions in Section 5.

2. A Language Engineering Fingerprint

Although there is a broad range of language statistics available, we only use a carefully handpicked set. We believe this set should be limited to support direct comparisons within one representative chart: a *language engineering fingerprint*. Furthermore, we only use language statistics, which can be easily and quickly calculated without the need for advanced language processing modules, e.g. part-of-speech (POS) taggers or syntax parsers. Such modules are usually highly text type-dependent (Sekine, 1997) and hence cannot be directly applied to previously unknown text types, as the selection of the most appropriate modules is precisely the goal of the analysis.

1. Shannon's *entropy* H measures the average amount of information in an underlying data structure. Applied in the field of language engineering, the mean amount of information of a token t_i can be calculated by approximating its probability $p(t_i)$ via its frequency in a given corpus. The entropy as given in Formula 1 is normalized to the vocabulary size $|V|$, i.e. the number of types in the corpus:

$$H = -\sum_{t_i \in V} p(t_i) \log_{|V|} p(t_i) \quad (1)$$

   A high entropy indicates that many words occur with small frequencies – instead of few words that occur with large frequencies.

2. The *relative vocabulary size* R_{Voc} (Těšitelová, 1992, chapter 1.2.3.3) is given by the ratio of the vocabulary size $|V|$ and the total number of tokens N_m with respect to "meaningful" words. These are defined as words, that are not function words ($N_m = \{t \mid t \notin$

$N_f\}$)[1], e.g. nouns, adjectives and verbs:

$$R_{\text{Voc}} = \frac{|V|}{N_m} \quad (2)$$

A small relative vocabulary size indicates simple language, less morphological constructs and few spelling and tokenization errors. Thus, it provides information about word repetition and the success of dictionary-based methods.

3. The *vocabulary concentration* C_{Voc} (Těšitelová, 1992, chapter 1.2.3.3) is defined by the ratio of the total number of tokens N_{top} with respect to the most frequent terms in the vocabulary V ($V_{top} = \{t \mid t \in V \wedge r(t) \leq 10\}$) and the total number of tokens N in a corpus

$$C_{\text{Voc}} = \frac{N_{top}}{N} \quad (3)$$

where rank $r(t)$ is defined as the position of a token t in a frequency-ordered list. A high vocabulary concentration indicates the corpus is made up of only a few words. Dictionary- and rule-based methods are then easier to implement and to maintain than for corpora with low vocabulary concentration.

4. The *vocabulary dispersion* D_{Voc} expresses the relative amount of low frequency tokens ($V_{\text{low}} = \{t \mid t \in V \wedge f(t) \leq 10\}$) in the vocabulary V:

$$D_{\text{Voc}} = \frac{|V_{\text{low}}|}{|V|} \quad (4)$$

where frequency $f(t)$ is defined as the number of occurrences of the token t in a corpus. A high vocabulary dispersion indicates a high fraction of spelling and tokenization errors or a morphologically rich language. Generally, this may significantly blow up language models and lead to large parameter spaces for machine learning methods. However, a high vocabulary dispersion may also give the opportunity (or obligation) to drastically apply pruning methods. Methods like POS tagging or named entity recognition (NER) are vulnerable to out-of-vocabulary words, which are more likely to occur in a corpus with a high vocabulary dispersion (Toutanova et al., 2003). If co-occurrences are to be calculated, significance measures like mutual information should be discarded for corpora with high vocabulary dispersion, as rare events would be overestimated.

5. The *corpus predictability* CP expresses the transition probabilities between tokens. For this, we need to calculate the entropy of a first-order Markov source $\mathcal{S}$ of two tokens t_i, t_j as given in Formula 5

$$H(\mathcal{S}) = -\sum_{t_i} p(t_i) \sum_{t_j} p_{t_i}(t_j) \log p_{t_i}(t_j) \quad (5)$$

where $p_{t_i}(t_j)$ denotes the probability of t_j given that it is preceded by t_i. CP is then calculated by normalizing the entropy of a first-order Markov source by its maximum possible entropy and subtracting it from 1:

$$CP = 1 - \frac{H(\mathcal{S})}{H_{max}(\mathcal{S})} \quad (6)$$

A high corpus predictability indicates a very straightforward writing style with words often followed by the same successors. This is an advantageous behavior for Hidden Markov Models (HMMs) or the calculation of neighborhood co-occurrences. If HMMs of higher order are used, the corpus predictability can be calculated accordingly.

6. A rudimentary *grammatical complexity* GC can be calculated by the ratio of the number of function words N_f to the number of meaningful words N_m:

$$GC = \frac{N_f}{N_m} \quad (7)$$

Although this rather basic approach cannot state a real level of grammatical structure in a corpus, it still provides evidence for the amount of effort put into expressing syntax. In conjunction with the average sentence length this language statistics may give guidance in which manner texts need to be processed, e.g. deep or shallow. For example, it may be possible to use rule-based POS taggers instead of more sophisticated POS taggers and thus significantly reduce calculation time. Sophisticated syntax parsers may be replaced by regular expression patterns (Trabold, 2007, chapter 8.2.3).

7. The *average sentence length* L_S influences parsing, relation extraction etc. The length $|s|$ of a sentence s is defined by the amount of tokens it contains, and the average sentence length of all sentences S is defined as in Formula 8:

$$L_S = \frac{1}{|S|} \sum_{s \in S} |s| \quad (8)$$

8. The *spelling accuracy* SA_{Voc} is defined by the amount of correctly spelled words N_{Cor} with respect to the total number of tokens N:

$$SA_{\text{Voc}} = \frac{N_{\text{Cor}}}{N} \quad (9)$$

This measure can furthermore be divided with respect to real-word errors and non-word errors and crucially influences which kind of spell-checking method needs to be employed, if any. A low spelling accuracy can significantly reduce the performance of context-based methods and machine learning in general.

9. Before developing an information extraction system, one should determine the corpus' *information density* ID_{Corp}, which is given by the ratio of relevant words N_r which are to be extracted and the total amount of tokens N:

$$ID_{\text{Corp}} = \frac{N_r}{N} \quad (10)$$

[1] As function words N_f we defined: the, a, an, he, him, she, her, they, us, we, them, it, his, to, on, above, below, before, from, in, for, after, of, with, at, and, or, but, nor, yet, so either, neither, both, whether

2.1. Further Language Statistics

Apart from the language statistics listed above, there are several more textual characteristics, which are tempting to use. Here, we list some of them and argue why we don't consider them appropriate to characterize corpora for language engineering purposes.

1. The product of a token's rank $r(t_i)$ and its frequency $f(t_i)$ is known to be approximately constant (Zipf, 1949), and is given by *Zipf's first law*:

$$r(t_i)f(t_i) \approx k \quad (11)$$

   Although this regularity does not always fit well (Melz and Wittig, 2007), we still do not consider the deviations from Zipf's first law large enough to provide much insight for language engineering.

2. Measures like the *Gunning-Fox-Index* (Gunning, 1952) are often used to assess the *readability* of text, for example to select texts appropriate for children of different ages. The Gunning-Fox-Index estimates how many years of education are needed to understand a given text. Although this measure is helpful for educational purposes, we do not consider it useful for language engineering, as it merely aggregates the average sentence length and the percentage of "complex" words containing three syllables or more, excluding compounds and proper nouns. While the amount of complex words may actually be a problem for a human reader, it can be argued that they do not necessarily pose difficulties for machine learning algorithms, as they can be treated as every other feature of a text.

3. Especially in technical genres and domain-specific language, words tend to be syntactically less ambiguous than in more general language, making it therefore easier to assign POS tags to them. This is reflected in the *syntactic ambiguity* A_S of a corpus. A_S is defined as the average entropy of a token t_i's POS tag s_j. A token occurring with only one POS tag has an entropy (and hence ambiguity) of zero, while its ambiguity increases for more syntactic classes being assigned to it with lesser probabilities:

$$A_S = -\frac{1}{|V|} \sum_{t_i} \sum_{s_j} p(s_j|t_i) \log p(s_j|t_i) \quad (12)$$

   A high syntactic ambiguity indicates possible difficulties for POS tagging, while a syntactic ambiguity stands for easy or even trivial POS tagging. Although we consider this measure as highly useful for language engineering, we discard it in the remainder, as it is very hard to calculate without advanced language processing modules.

4. Correspondingly to A_S, the *lexical ambiguity* A_L is defined by the average entropy of a token t_i's meaning m_j:

$$A_L = -\frac{1}{|V|} \sum_{t_i} \sum_{m_j} p(m_j|t_i) \log p(m_j|t_i) \quad (13)$$

   Intuitively, a high lexical ambiguity complicates information extraction massively, while corpora with only a few ambiguous words with few meanings are easier handle. Again, we discard this measure in the remainder, as there is no way to calculate it automatically.

2.2. General Remarks on Language Statistics

Although we avoided language statistics which can only be calculated using advanced language processing methods, we consider it feasible to inspect small samples of text manually. For example, the spelling accuracy can be semi-manually estimated from several hundred words.

Unfortunately, most language statistics cannot easily be normalized by corpus size and thus should be calculated on same- or at least similar-sized corpora. Obviously, language statistics may not be accurate enough on samples smaller than several ten thousand words.

Finally, the language statistics selected and presented by us cannot easily be interpreted as being in some way "better" or "worse" for higher or lower values when comparing corpora. An interpretation is always bound to the nature of the task to be accomplished. For the same reason, it is hard or even impossible to combine all textual characteristics into one artificial "quality" measure of some sort.

3. Language Engineering Fingerprints of Corpora from Different Genres

We now apply the textual characteristics described in Section 2. to three English-language corpora from different genres: English-language news articles from *WikiNews*[2], posts to automotive web fora, and automotive repair orders. News articles were chosen as much scientific work focuses on then, posts to web fora are of increasing importance for web mining, and repair orders, which are scientifically not covered yet, are highly relevant for many manufacturing companies. Furthermore, news articles, posts to web fora and repair orders reflect an increasing level of *genre-specific language*. As textual characteristics are partly influenced by corpus size, language statistics are calculated on three million token samples belonging to randomly chosen sentences for each corpus.

In Figure 1 the increasing level of genre-specific language becomes evident. Relative vocabulary size is very small for repair orders, reflecting the restricted and highly genre-specific language. The high vocabulary dispersion in repair orders is an indication for a low spelling accuracy. Interestingly, the level of grammatical complexity and average sentence length decreases greatly from news to web fora to repair orders; repair orders tend to be expressed in short and simply structured phrases. A surprising outcome is the small vocabulary concentration of repair orders, which is explained by the low frequency of function words. While the top 10 most frequent words of news and posts to web fora mainly include function words, the most frequent words of repair orders also include meaningful "content" words, which are also expressed by synonyms. Repair order's vocabulary concentration is higher if calculated using the top 100 most frequent words instead of the top 10.

[2] http://en.wikinews.org

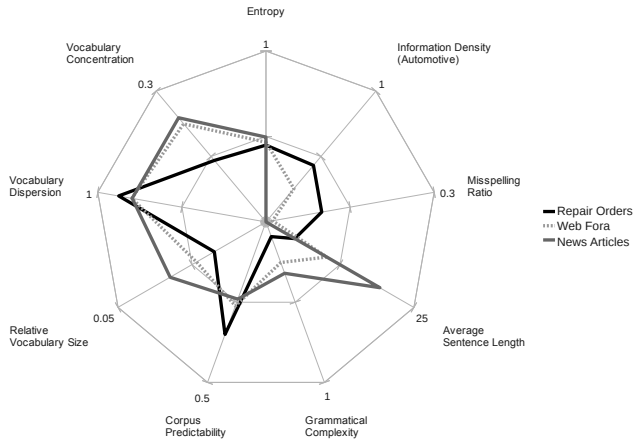

Figure 1: Language engineering fingerprints of news articles, posts to web fora and repair orders. Misspelling Ratio is defined as $1 - SA_{\text{Voc}}$.

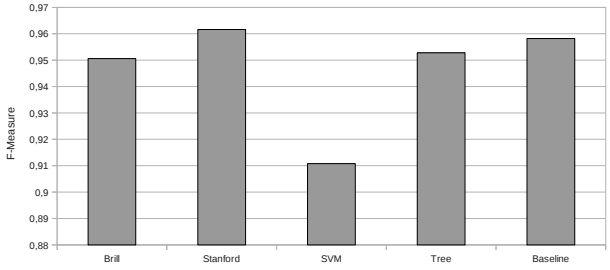

Figure 2: POS tagging results (F-Measures) on repair orders using Brill tagger (Brill, 1992), Stanford POS tagger (Toutanova et al., 2003), Support Vector Machines (Giménez and Márquez, 2004) and decision trees (Schmid, 1994) compared to a naive baseline.

4. A Case Study: Information Extraction from Automotive Repair Orders

After examining the language engineering fingerprints of news articles, posts to web fora and repair orders, we carry out a case study to demonstrate how textual characteristics may give guidance to select appropriate algorithms for a successful information extraction system on repair orders (cf. Figure 1). Their high vocabulary dispersion can be explained easily upon inspection of several hundred words: repair orders contain a large amount of tokenization errors, misspellings and technical codes and hence require an adapted pre-processing and tokenization. The small relative vocabulary size and the high corpus predictability suggests to employ a dictionary-based spelling correction using neighborhood co-occurrences (Schierle et al., 2008). This genre-specific method indeed yields good results in comparison to other methods.

After pre-processing, tokenization and spelling correction we POS tag our repair orders. Different state-of-the art methods were evaluated; evaluation results are shown in in Figure 2. Trabold (2007) presents details regarding the evaluation and discusses its results. Remarkably, only the Stanford POS tagger (Toutanova et al., 2003) outperforms the naive baseline of simply tagging every word with its most frequent tag. This is due to a low syntactic ambiguity, and the small amount of undetected tokenization errors, misspellings and out-of-vocabulary words.

Finally, we extract relations between components, failure symptoms, their corrections etc. On the one hand, preliminary tests using the Stanford parser (Klein and Manning, 2003a; Klein and Manning, 2003b) showed, repair orders – characterized by low grammatical complexity – cannot easily be parsed by an off-the-shelf syntax parser (Hormazábal, 2007). On the other hand, re-training existing methods or creating a special grammar is very time consuming. However, given the low grammatical complexity, high corpus predictability and low syntactic ambiguity of repair orders, it is reasonable to rely on unsupervised methods for both POS tagging and syntax parsing. Therefore, we incorporated UnsuPOS (Biemann, 2006) and UnsuParse (Hänig et al., 2008) as a basis for our relation extraction. Although they cannot easily be evaluated directly or compared to supervised methods, their usefulness was assessed indirectly via the performance of our rule-based relation extraction step (Hänig and Schierle, 2009): we reach an F-Measure of partly more than 0.90, which allows us to conclude that UnsuPOS and UnsuParse yield satisfying results. Schierle (2011) provides an overview of the whole system as well as more detailed evaluations etc.

In summary, we built a system to extract information from automotive repair orders which exhibits highly genre- and domain-specific language. Thereby, we achieved very good results, not necessarily because of the superiority of one or more algorithms, but merely because each module of the system was carefully tailored to the genre's textual characteristics.

5. Conclusion

Over the years, research in NLP has led to a plethora of methods and algorithms, often encapsulated in modules such as tokenization, sentence segmentation, POS tagging, syntax parsing, NER, relation extraction etc. For each module there is wide variety of approaches; POS tagging alone was approached using rules (Brill, 1992), HMMs (Brants, 2000), Support Vector Machines (Giménez and Márquez, 2004), decision trees (Schmid, 1994) etc. In turn, each approach may be evaluated on a wide variety of corpora. Obviously, it is hard if not impossible to compile and maintain a comprehensive overview of all approaches, their evaluations and textual characteristics of the respective corpora. Therefore, we hope to encourage other researchers to follow our endeavor and describe corpora they work on using language statistics and hence facilitate comparability, reproducibility and transferability of their methods and results. In the light of social media, where comparatively "new" genres like chat protocols, posts to blogs and fora, tweets, wikis etc. are likely to all have a completely different textual characteristics, this becomes of increasing importance.

In this paper, we selected and presented a set of suitable language statistics. Its intended use is to compare corpora

originating from different genres and domains and to estimate the transferability of NLP methods and algorithms from one corpus to another. We do not consider this set to be complete or perfectly adequate for every language engineering process, but we appeal to the scientific community to contribute to it and to provide feedback, so that over time a standard can be established. We measured textual characteristics on corpora from three different genres and derived their language engineering fingerprint. In a case study we have shown, how textual characteristics may give guide to select appropriate algorithms for a successful information extraction system on a highly specific genre: automotive repair orders.

6. References

Chris Biemann. 2006. Unsupervised Part-of-Speech Tagging Employing Efficient Graph Clustering. In *Proceedings of the 21st International Conference on Computational Linguistics (COLING) / Association for Computational Linguistics (ACL) Student Research Workshop*, pages 7–12.

Thorsten Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proceedings of the 6th Conference Applied Natural Language Processing (ANLP)*, pages 224–231.

Eric Brill. 1992. A simple rule-based part of speech tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing (ANLP)*, pages 112–116.

Hamish Cunningham. 1999. A definition and short history of language engineering. *Natural Language Engineering*, 5:1–16, March.

Gerard Escudero, Lluís Màrquez, and German Rigau. 2000. An empirical study of the domain dependence of supervised word sense disambiguation systems. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC)*, pages 172–180.

Jess Giménez and Lluís Márquez. 2004. SVMTool: A general POS tagger generator based on support vector machines. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, pages 43–46.

Robert Gunning. 1952. *The Technique of Clear Writing*. McGraw-Hill, New York.

Christian Hänig and Martin Schierle. 2009. Relation extraction based on unsupervised syntactic parsing. In Gerhard Heyer, editor, *Proceedings of the Conference on Text Mining Services (TMS)*, pages 65–70.

Christian Hänig, Stefan Bordag, and Uwe Quasthoff. 2008. UnsuParse: Unsupervised parsing with unsupervised part of speech tagging. In *Proceedings of the 6th International Language Resources and Evaluation (LREC)*, pages 1109–1114.

David Holmes. 1994. Authorship attribution. *Computers and the Humanities*, 28:87–106.

Miguel Hormazábal. 2007. Implemation of a factored parser for the automatic classification of customer reports. Master's thesis, Eberhard Karls University Tübingen.

Dan Klein and Christopher D. Manning. 2003a. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 423–430.

Dan Klein and Christopher D. Manning. 2003b. Fast exact inference with a factored model for natural language parsing. *Advances in Neural Information Processing Systems*, 15:3–10.

Ronny Melz and Thomas Wittig. 2007. Universal regularities of language statistics. In *Statistical Physics of Social Dynamics: Opinions, Semiotic Dynamics, and Laguage, Satellite Workshop of STATPHYS 2007*.

Martin Schierle, Sascha Schulz, and Markus Ackermann. 2008. From spelling correction to text cleaning – using context information. *Data Analysis, Machine Learning and Applications*, pages 397–404.

Martin Schierle. 2011. *Language Engineering for Information Extraction*. Ph.D. thesis, Leipzig University.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing (ICNLP)*, pages 44–49.

Satoshi Sekine. 1997. The Domain Dependence of Parsing. In *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP)*, pages 96–102.

Marie Těšitelová. 1992. *Quantitive Linguistics*. John Benjamins Publishing Company, Amsterdam/Philadelphia.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the Human Language Technologies: North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 173–180.

Daniel Trabold. 2007. Konzeption und Realisierung eines multilingualen Systems zur Erkennung benannter Entitten. Master's thesis, Leipzig University.

Dong Wang and Yang Liu. 2011. A cross-corpus study of unsupervised subjectivity identification based on calibrated em. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA)*, pages 161–167.

George K. Zipf. 1949. Human behaviour and the principle of least-effort. Addison-Wesley, Cambridge, MA.