On the Correspondence between Compositional Matrix-Space Models of Language and Weighted Automata

Shima Asaadi* and **Sebastian Rudolph**
Faculty of Computer Science
Technische Universität Dresden
firstname.lastname@tu-dresden.de

Abstract

Compositional matrix-space models of language were recently proposed for the task of meaning representation of complex text structures in natural language processing. These models have been shown to be a theoretically elegant way to model compositionality in natural language. However, in practical cases, appropriate methods are required to learn such models by automatically acquiring the necessary token-to-matrix assignments. In this paper, we introduce graded matrix grammars of natural language, a variant of the matrix grammars proposed by Rudolph and Giesbrecht (2010), and show a close correspondence between this matrix-space model and weighted finite automata. We conclude that the problem of learning compositional matrix-space models can be mapped to the problem of learning weighted finite automata over the real numbers.

1 Introduction

Quantitative models of language have recently received considerable research attention in the field of Natural Language Processing (NLP). In the application of meaning representation of text in NLP, much effort has been spent on semantic Vector Space Models (VSMs). Such models capture word meanings quantitatively, based on their statistical co-occurrences in the documents. The basic idea is to represent words as vectors in a high-dimensional space, where each dimension corresponds to a separate feature. In this way, semantic similarities can be computed based on measuring the distance between vectors in the vector space (Mitchell and Lapata, 2010). Vectors which are close together in this space have similar meanings and vectors which are far away are distant in meaning (Turney and Pantel, 2010).

VSMs typically represent each word separately, without considering representations of phrases or sentences. So, the compositionality properties of the language is lost in VSMs (Mitchell and Lapata, 2010). Recently, some approaches have been developed in the area of compositionality and distributional semantics in NLP. These approaches introduce different word representations and ways of combining those words. Mitchell and Lapata (2010) propose a framework for vector-based semantic composition. They define additive or multiplicative function for the composition of two vectors and show that compositional approaches generally outperform non-compositional approaches which treat the phrase as the union of single lexical items. However, VSMs still have some limitations in the task of modeling complex conceptual text structures. For example, in the bag-of-words model, the words order and therefore the structure of the language is lost.

To overcome the limitations of VSMs, Rudolph and Giesbrecht (2010) proposed Compositional Matrix-Space Models (CMSM) as a recent alternative model to work with distributional approaches. These models employ matrices instead of vectors and make use of iterated matrix multiplication as the only composition operation. They show that these models are powerful enough to subsume many known models, both quantitative (vector-space models with diverse composition operations) and qualitative ones (such as regular languages). It is also proved theoretically that this framework is an elegant way to model compositional, symbolic and distributional aspects of natural language.

However, in practical cases, methods are needed

*Supported by DFG Graduiertenkolleg 1763 (QuantLA)

to automatically acquire the token-to-matrix assignments from available data. Therefore, methods for training such models should be developed e.g. by leveraging appropriate machine learning methods.

In this paper, we are concerned with Graded Matrix Grammars, a variant of the Matrix Grammars of Rudolph and Giesbrecht (2010), where instead of the "yes or no" decision, if a sequence is part of a language, a real-valued score is assigned. This is a popular task in NLP, used, e.g., in sentiment analysis settings (Yessenalina and Cardie, 2011).

Generally, in many tasks of NLP, we need to estimate functions which map arbitrary sequence of words (e.g. sentences) to some semantical space. Using Weighted Finite Automata (WFA), an extensive class of these functions can be defined, which assign values to these sequences (Balle and Mohri, 2012).

Herein, inspired by the definition of weighted finite automata (Sakarovitch, 2009) and their applications in NLP (Knight and May, 2009), we show a tight correspondence between graded matrix grammars and weighted finite automata. Hence, we argue that the problem of learning CMSMs can be mapped to the problem of learning WFA.

The rest of the paper is organized as follows. Section 2 provides the basic notions of weighted automata and the matrix-space model. A detailed description of correspondence between CMSM and WFA is presented in Section 3, followed by related work in Section 4 and conclusion and future work in Section 5.

2 Preliminaries

In this section, we provide the definitions of weighted automata in (Balle and Mohri, 2015; Sakarovitch, 2009) and matrix-space models of language in (Rudolph and Giesbrecht, 2010).

2.1 Weighted Finite Automata

Weighted finite automata generalize classical automata in which transitions and states carry weights. These weights can be considered as the cost of the transitions or amount of resources needed to execute the transitions. Let Σ be a finite alphabet. A weighted automaton $\mathcal{A}$ is a tuple of $(Q_\mathcal{A}, \lambda, \alpha, \beta)$ and defined over a semi-ring $(\mathbb{S}, \oplus, \otimes, \bar{0}, \bar{1})$. $Q_\mathcal{A}$ is a finite set of states, $\lambda : \Sigma \to \mathbb{S}^{Q_\mathcal{A} \times Q_\mathcal{A}}$ is the transition weight function,

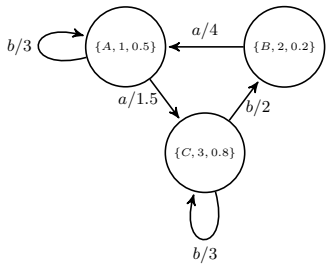

Figure 1: Example of WFA $\mathcal{A}$.

and, $\alpha : \Sigma \to \mathbb{S}$ and $\beta : \Sigma \to \mathbb{S}$ are two functions assigning to every state its initial and final weight. Thereby, for each transition $e = (q, \sigma, q')$, $\lambda(\sigma)_{q,q'}$ denotes the weight of the label σ associated with the transition e between q and q', which are the source and target state of the transition. Moreover, A *path* $\mathcal{P}$ in $\mathcal{A}$ is a sequence of transitions labeled with $\sigma_1 \cdots \sigma_n$, in more detail:

$$\mathcal{P} := p_0 \xrightarrow{\sigma_1} p_1 \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_n} p_n$$

with $p_i \in Q_\mathcal{A}$. The *weight* of $\mathcal{P}$ is defined as the $\otimes$-product of the weights of the starting state, its transitions, and final state: $\omega(\mathcal{P}) = \alpha(p_0) \otimes \lambda(\sigma_1)_{p_0,p_1} \otimes \cdots \otimes \lambda(\sigma_n)_{p_{n-1},p_n} \otimes \beta(p_n)$. Now, the weight of a word $x = \sigma_1 \cdots \sigma_n \in \Sigma^\star$ is the cumulative weight of all paths labeled with the sequence $\sigma_1 \cdots \sigma_n$ which is computed as the $\oplus$-sum of the weights of the corresponding paths, also known as a *rational power series*:

$$f_\mathcal{A}(\sigma_1 \cdots \sigma_n) = \bigoplus_{\mathcal{P} \in P_\mathcal{A}(\sigma_1 \cdots \sigma_n)} \omega(\mathcal{P}), \quad (1)$$

where $P_\mathcal{A}(\sigma_1 \cdots \sigma_n)$ denotes the (finite) set of paths in $\mathcal{A}$ labeled with $\sigma_1 \cdots \sigma_n$. So, the function $f_\mathcal{A}$ maps the set of strings in $\Sigma^\star$ to $\mathbb{S}$. In this work, we will assume that $\mathbb{S}$ is the set of the real numbers $\mathbb{R}$ with the usual multiplication and addition. Figure 1 illustrates an example of WFA over $\Sigma = \{a, b\}$. Inside each state there is a tuple of the name, initial and final weight of the state, respectively. As an example, for $x = ab$ we have: $f_\mathcal{A}(x) = 1 \times 1.5 \times 3 \times 0.8 + 1 \times 1.5 \times 2 \times 0.2 + 2 \times 4 \times 3 \times 0.5$.

2.2 Compositionality and Compositional Matrix-Space Model

The general principle of compositionality is that the meaning of a complex expression is a function of the meaning of its constituent tokens and some rules used to combine them (Frege, 1884). More

formally, according to Rudolph and Giesbrecht (2010), the underlying idea can be described as follows: "Given a mapping $[\![\cdot]\!] : \Sigma \to \mathbb{S}$ from a set of tokens in Σ into some semantical space $\mathbb{S}$, the composition operation is defined by mapping sequences of meanings to meanings: $\bowtie$: $\mathbb{S}^\star \to \mathbb{S}$. So, the meaning of the sequence of tokens $\sigma_1 \cdots \sigma_n$ can be obtained by first applying the function $[\![\cdot]\!]$ to each token and then $\bowtie$ to the sequence $[\![\sigma_1]\!] \cdots [\![\sigma_n]\!]$, as shown in Figure 2".

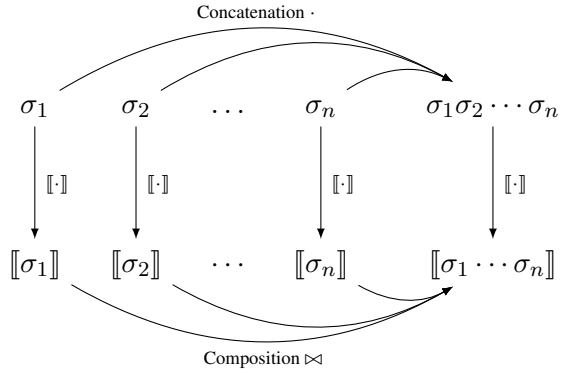

Figure 2: Principle of compositionality, illustration taken from Rudolph and Giesbrecht (2010)

In compositional matrix-space models, this general idea is instantiated as follows: we have $\mathbb{S} = \mathbb{R}^{n\times n}$, i.e., the semantical space consists of quadratic matrices of real numbers. The mapping function $[\![\cdot]\!]$ maps the tokens into matrices so that the semantics of simple tokens is expressed by matrices. Then, using the standard matrix multiplication as the only composition operation $\bowtie$, the semantics of complex phrases are also described by matrices.

Rudolph and Giesbrecht (2010) showed theoretically that by employing matrices instead of vectors, CMSMs subsume a wide range of linguistic models such as statistical models (vector-space models and word space models).

3 Graded Matrix Grammars and Weighted Finite Automata

In some applications of NLP, we need to derive the meaning of a sequence of words in a language, which can be done with CMSMs as described in Section 2.2. In this section, we introduce the notion of a graded matrix grammar which constitutes a slight variation of matrix grammars as introduced by Rudolph and Giesbrecht (2010).

Definition 1 (Graded Matrix Grammars)**.** *Let Σ be an alphabet. A graded matrix grammar $\mathcal{M}$ of degree n is defined as the tuple $\langle [\![\cdot]\!], \Sigma, \alpha, \beta\rangle$ where $[\![\cdot]\!]$ is a function mapping tokens in Σ to $n \times n$ matrices of real numbers. Moreover, $\alpha, \beta \in \mathbb{R}^n$. Then we map each sequence of tokens $\sigma_1 \cdots \sigma_n \in \Sigma^\star$ to a real number (called the value of the sequence) using the target function $\varphi : \Sigma^\star \to \mathbb{R}$ defined by:*

$$\varphi(\sigma_1 \cdots \sigma_n) = \alpha[\![\sigma_1]\!] \cdots [\![\sigma_n]\!]\beta^\top. \qquad (2)$$

However, as discussed before, to be used in practice, some learning methods are needed to extract graded matrix grammars from textual data. Hence, the target function φ can be generalized to all texts in the language and handle unseen word compositions. To this end, we show the correspondence between the CMSM and WFA, with the consequence that existing learning methods for WFA can be applied to learning CMSMs.

As discussed in Section 2.1, in WFA, for a rational power series f, a value $f(x)$ is the sum of all possible paths labeled with $x = \sigma_1 \cdots \sigma_n \in \Sigma^\star$. However, this computation can be described via matrices by the fact that a walk over a graph corresponds to a matrix multiplication (Sakarovitch, 2009). More precisely, for any $\sigma \in \Sigma$, let $A_\sigma \in \mathbb{R}^{Q_\mathcal{A}\times Q_\mathcal{A}}$ be the transition matrix of σ: $[A_\sigma]_{pq} = \sum_{e\in P_\mathcal{A}(p,\sigma,q)} \lambda(\sigma)_{p,q}$, where $P_\mathcal{A}(p,\sigma,q)$ is the set of all transitions labeled with σ from p to q. Also, the vectors $\alpha_\mathcal{A} \in \mathbb{R}^{Q_\mathcal{A}}$ and $\beta_\mathcal{A} \in \mathbb{R}^{Q_\mathcal{A}}$ are the start and final weights of the states in $Q_\mathcal{A}$, respectively. Then, Equation 1 can be equally expressed as follows in terms of matrices with entries in $\mathbb{R}$ (Balle and Mohri, 2015):

$$f_\mathcal{A}(\sigma_1 \cdots \sigma_n) = \alpha_\mathcal{A}^\top A_{\sigma_1} \cdots A_{\sigma_n}\beta_\mathcal{A} \qquad (3)$$

Hence, we see the correspondence between Equation 2 and 3. In more detail, consider each phrase p and its value r in a natural language. If we extract the words of the language as a finite alphabet Σ in an automaton, then p would be a string in $\Sigma^\star$. The $[\![\cdot]\!]$ function in $\mathcal{M}$ applied over the words constructs $n\times n$ transition matrices of the alphabets in the automaton. Here, n can be the number of states of the automaton. So, estimating the function φ in graded matrix grammar corresponds to estimating the target function of the automaton $f_\mathcal{A}$, which computes exactly the value of a string translated from the phrase p in a language. That is, the representation of a string is done with multiplication of transition matrices of its tokens, which results in a

new representation matrix for the string. Then, the suitable predefined vectors α and β translate the resulting matrix to a real number which denotes the value of associated phrase p in the natural language.

The problem of learning WFAs is finding a WFA closely estimating a target function, using for training a finite sample of strings labeled with their target values (Balle and Mohri, 2015). By learning WFAs, one obtains an automaton that is a tuple $\mathcal{A} = \langle \alpha, \beta, \{A_a\}_{a \in \Sigma} \rangle$, and one can compute the target function $f_{\mathcal{A}}(x)$. Since WFA encode CMSMs, and based on this close correspondence between them, learning a graded matrix grammar to estimate the value of phrases can be mapped to the problem of learning a weighted automaton.

4 Related Work

An application of CMSM has been shown in the work of Yessenalina and Cardie (2011). They proposed a learning-based approach for phrase-level sentiment analysis. Inspired by the work of Rudolph and Giesbrecht (2010) they use CMSMs to model composition, and present an algorithm for learning a matrix for each word via ordered logistic regression, which is evaluated with promising results. However, it is not trivial to learn a matrix-space model. Since the final optimization problem is non-convex, the matrix initialization for this method is not done perfectly.

Socher et al. (2012) introduce a matrix-vector recursive neural network (MV-RNN) model that learns compositional vector representations for phrases and sentences. The model assigns a vector and a matrix to every node in a parse tree. The vector represents the meaning of the constituent, while the matrix captures how it affects the meaning of neighboring constituent. The model needs to parse the tree to learn the vectors and matrices.

Recently, new approaches are proposed in learning weighted finite automata in NLP. Balle and Mohri (2012) and Balle et al. (2014) introduce a new family of algorithms for learning general WFA and stochastic WFA based on the combination of matrix completion problem and spectral methods. These algorithms are designed for learning an arbitrary weighted automaton from sample data of strings and assigned labels. They formulate the missing information from the sample data as a Hankel matrix completion problem. Then, the spectral learning is applied to the resulting Hankel matrix to obtain WFA. Balle et al. (2014) also, offer the main results in spectral learning which are an interesting alternative to the classical EM algorithms in the context of grammatical inference and show the computational efficiency of these algorithms.

Moreover, Balle and Mohri (2015) discuss modern learning methods (spectral methods) for an arbitrary WFA in different scenarios. They provide WFA reconstruction algorithms and standardization. it is theoretically guaranteed that for a Hankel matrix with a finite rank, representing a rational power series, there is a corresponding WFA with the number of states equal to this rank and it is minimal.

5 Conclusion and Future Work

In this paper, we introduced a graded matrix grammar for compositionality in language where compositional matrix-space models are employed in different tasks of NLP. However, we need to propose a learning method to train this model for value assignments in NLP. For this purpose, we showed the close correspondence between matrix grammars and weighted automata. So, the problem of learning the CMSM can be encoded as the problem of learning WFA.

Our future goal is to review the existing methods in learning WFA, and adapt them to solve the task of sentiment analysis/meaning representation in NLP. Using learning methods, they allow to automatically learn CMSM and induce the graded matrix grammar.

References

Borja Balle and Mehryar Mohri. 2012. Spectral learning of general weighted automata via constrained matrix completion. In *Advances in neural information processing systems*, pages 2168–2176.

Borja Balle and Mehryar Mohri, 2015. *Learning Weighted Automata*, pages 1–21. Springer International Publishing.

Borja Balle, Xavier Carreras, Franco M Luque, and Ariadna Quattoni. 2014. Spectral learning of weighted automata. *Machine Learning*, 96(1):33–63.

Gottlob Frege. 1884. *Die Grundlagen der Arithmetik eine logisch-mathematische Untersuchung über den Begriff der Zahl*. Verlage Wilhelm Koebner, Breslau.

Kevin Knight and Jonathan May. 2009. Applications of weighted automata in natural language processing. In Manfred Droste, Werner Kuich, and Heiko Vogler, editors, *Handbook of Weighted Automata*, pages 571–596. Springer Berlin Heidelberg, Berlin, Heidelberg.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

Sebastian Rudolph and Eugenie Giesbrecht. 2010. Compositional matrix-space models of language. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 907–916, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jacques Sakarovitch. 2009. Rational and recognisable power series. In Manfred Droste, Werner Kuich, and Heiko Vogler, editors, *Handbook of Weighted Automata*, pages 105–174. Springer Berlin Heidelberg, Berlin, Heidelberg.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. EMNLP-CoNLL '12, pages 1201–1211, Stroudsburg, PA, USA. Association for Computational Linguistics.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.

Ainur Yessenalina and Claire Cardie. 2011. Compositional matrix-space models for sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 172–182, Stroudsburg, PA, USA. Association for Computational Linguistics.