Acta Cybernetica **19** (2009) 313–356.

Statistical Language Models within the Algebra of Weighted Rational Languages

Thomas Hanneforth[*] and Kay-Michael Würzner[*]

Abstract

Statistical language models are an important tool in natural language processing. They represent prior knowledge about a certain language which is usually gained from a set of samples called a *corpus*. In this paper, we present a novel way of creating N-gram language models using weighted finite automata. The construction of these models is formalised within the algebra underlying weighted finite automata and expressed in terms of weighted rational languages and transductions. Besides the algebra we make use of five special constant weighted transductions which rely only on the alphabet and the model parameter N. In addition, we discuss efficient implementations of these transductions in terms of *virtual constructions*.

Keywords: computational linguistics, weighted rational transductions, statistical language modeling, N-gram models, weighted finite-state automata

1 Introduction

Weighted finite-state acceptors (WFSA) provide a convenient way to compactly represent N-gram language models (cf. [3]) since they admit equivalence transformations like determinisation and minimisation [22] which compress common prefixes and suffixes without changing the counts or probabilities associated with an individual N-gram. Moreover, it is possible to represent all sub-distributions of M-grams (with $1 \leq M < N$) simultaneously with almost no additional space.

The usual way is to construct the language models on the basis of the manipulation of states and transitions. Since the models are also required to be robust, it is necessary to reserve some probability mass for unseen N-grams. This is commonly achieved by combining a discounting method with a back-off [17] or interpolation mechanism [15]. The adjusted probabilities are then reassigned for each N-gram to existing or newly created transitions. The finite automata thus merely serve as a data structure.

[*] University of Potsdam, E-mail: {tom,wuerzner}@ling.uni-potsdam.de

In this paper, we present an approach which treats the creation of N-gram models as a problem of modifying weighted languages rather than states and transitions. In particular, we only use operations from the algebra of weighted regular languages (WRLs) and transductions (WRTs) like union and intersection to get from a set of samples to a robust back-off model. Such an algebraic formalisation has – at least to our knowledge – never been done before.

The results outlined in the remainder are by now mainly of theoretical interest. We do not aim to replace the many excellent statistical toolkits by the machinery proposed here. This work is rather a "case study" in viewing an important tool in natural languages processing from a theoretical viewpoint. As such, we describe it in a self-contained form.

This article is organised as follows: In Section 2, we will recall the notion of language models in general and N-gram models in particular (may be skipped by readers familiar with the topic). Section 3 introduces the formal preliminaries and establishes the notation. The subsequent sections 4-7 deal with the creation of N-gram and back-off models from scratch in the manner explained above. Matters of complexity and implementation are discussed in each section. Proofs of correctness of the outlined methods have been put in the appendix for reasons of readability.

2 Language Models

Language modeling is the task of assigning a probability to sequences of words. $\mathrm{Pr}(w)$ is the prior probability of the sequence of words w. Language models are used in many applications in natural language processing such as speech recognition, machine translation, optical character recognition or part-of-speech tagging. See [16] for an introduction to these topics and their relation to language models.

Using *conditional probabilities*, the joint probability of a sequence of words can be decomposed as: [1]

$$\mathrm{Pr}(w_1^m) = \mathrm{Pr}(w_1) \prod_{i=2}^{m} \mathrm{Pr}(w_i | w_1^{i-1}) \ . \tag{1}$$

The interdependencies of words are reflected by assuming that the occurrence of a word is a consequence of the occurrence of its predecessors. The conditional probability of a sequence of words can be computed by normalising its frequency relative to the frequency of its history ($\mathrm{C}(s)$ denotes the number of occurrences of a substring s in w, Σ refers to a finite alphabet and the sum operator, respectively):

$$\mathrm{Pr}(w_i | w_1^{i-1}) = \frac{\mathrm{C}(w_1^{i-1} \cdot w_i)}{\sum\limits_{a \in \Sigma} \mathrm{C}(w_1^{i-1} \cdot a)} \ . \tag{2}$$

[1] We denote a substring $w_i \dots w_j$ with $j \geq i$ in a more compact way by w_i^j. If $i = j$, we omit the superscript and write simply w_i for the i^{th} character of w (starting at 1). If the subscript exceeds the superscript, we implicitly denote the empty string ε.

Frequency information is obtained from a *(text) corpus* which is usually defined as a large collection of (annotated or unannotated) texts. In the remainder of this article, we use the term *corpus* as denoting a finite disjunction of *sentences*. A sentence roughly corresponds to its linguistic definition, but is not limited to that. Some big natural language corpora are the DWDS-Korpus [11] or the Brown corpus [18]. A *probability distribution* with respect to Σ^* is an assignment of probabilities to the strings in Σ^* such that all the individual string probabilities sum up to one.

2.1 N-Gram Models

N-gram models are the most widely used type of language models. Their success is based in their simplicity: They can be derived unsupervised, by just counting sequences of words in a corpus and computing their relative frequency.

In the field of language modeling, an N-gram is a sequence of N elements taken from a fixed and finite alphabet Σ, for example letters [29], words [3], morphemes, etc.

In order to limit the number of possible contexts of a word, it is assumed that sequences of words form *Markov chains* [20]. Thus, only the last $N-1$ words (sometimes also called the *history* of w_i) affect the word w_i:

$$\Pr(w_i|w_1^{i-1}) \approx \Pr(w_i|w_{i-(N-1)}^{i-1}) \; . \tag{3}$$

The number of possible contexts is then the size of the alphabet to the power of $N-1$ and therefore finite. The boundary case at the beginning of the sentence is handled by $N-1$ beginning-of-sentence markers (see Section 6 for details).

2.2 Smoothing

While theoretically possible, one will never find all potential N-grams in a corpus in practice. The common solution to this problem is *smoothing*: Probability mass is assigned to unseen events and/or other distributions which account for those events are consulted. For N-gram models, this means to change the model in such a way that it assigns a probability to any combination of N words of the vocabulary, deals adequately with out-of-vocabulary items and, is still a *probabilistic* model.

Probabilistic N-gram models are characterised by the property that for every context the probabilities of possible continuations sum up to one ($h \in \Sigma^{N-1}$):

$$\forall h \sum_{w_i} \Pr(w_i|h) = 1 \; . \tag{4}$$

Many different smoothing methods for different purposes are available (cf. [6] for a detailed summary and comparison of important smoothing methods).

For the purpose of this work, we recall the notions of *discounting* and *back-off* smoothing.

2.2.1 Discounting

The main idea behind this class of procedures is to redistribute probability mass from seen to unseen events. A simple but effective discounting algorithm is the so called *Witten-Bell discounting*, referring to method C in [30]. Witten-Bell discounting is based on the intuition that the probability of novel events decreases with the number of different events that are observed in the corpus. To implement this idea, the frequencies of the N-grams are normalised by the number of different N-grams sharing the same $(N-1)$-gram prefix. The number of different events in an event space is often called the number of *types*.

Definition 1 (Witten-Bell Type Number)**.** *Let* T *be a function* $\Sigma^* \to \mathbb{N}$:

$$\mathrm{T}(w_{i-N+1}^{i}) = \sum_{a \in \Sigma, \mathrm{C}(w_{i-N+1}^{i-1} \cdot a) \neq 0} 1 \ .$$

Definition 2 (Witten-Bell Token Number)**.** *Let* N *be a function* $\Sigma^* \to \mathbb{N}$:

$$\mathrm{N}(w_{i-N+1}^{i}) = \sum_{a \in \Sigma} \mathrm{C}(w_{i-N+1}^{i-1} \cdot a) \ .$$

With the help of the functions T and N it is possible to discount frequencies, denoted by $\tilde{\mathrm{C}}$:

$$\tilde{\mathrm{C}}(w_{i-N+1}^{i}) = \mathrm{C}(w_{i-N+1}^{i}) \frac{\mathrm{N}(w_{i-N+1}^{i})}{\mathrm{N}(w_{i-N+1}^{i}) + \mathrm{T}(w_{i-N+1}^{i})} \ . \tag{5}$$

Adjusted probabilities $\tilde{\mathrm{Pr}}$ can be computed from $\tilde{\mathrm{C}}$ [16]. The freed frequency mass is computed by:

$$\begin{aligned} &\sum_{w_{i-N+1}^{i} \in \Sigma^N} \mathrm{C}(w_{i-N+1}^{i}) - \tilde{\mathrm{C}}(w_{i-N+1}^{i}) \\ = &\sum_{w_{i-N+1}^{i} \in \Sigma^N} \mathrm{C}(w_{i-N+1}^{i}) \ \frac{\mathrm{T}(w_{i-N+1}^{i})}{\mathrm{N}(w_{i-N+1}^{i}) + \mathrm{T}(w_{i-N+1}^{i})} \ . \end{aligned} \tag{6}$$

2.2.2 Smoothing by Combining Different Distributions

Spreading saved probability mass equally among all unseen events is often too simple. It seems reasonable to take different distributions into account. A common way of doing that is the *back-off* strategy [17] which recursively uses the $(N-1)$-gram distribution whenever the N-gram distribution assigns a zero probability. Equation (7) formalises this behavior by defining the *back-off probability* $\hat{\mathrm{Pr}}$:

$$\begin{aligned} \hat{\mathrm{Pr}}(w_i | w_{i-N+1}^{i-1}) = &\ \tilde{\mathrm{Pr}}(w_i | w_{i-N+1}^{i-1}) \\ &+ \phi(\tilde{\mathrm{Pr}}(w_i | w_{i-N+1}^{i-1})) \\ &\cdot \alpha(w_{i-N+1}^{i-1}) \hat{\mathrm{Pr}}(w_i | w_{i-N+2}^{i-1}) \ . \end{aligned} \tag{7}$$

Function ϕ indicates the need for backing-off to the immediately lower ordered distribution:

$$\phi(x) = \begin{cases} 0 & \text{if } x \neq 0 \\ 1 & \text{otherwise} \ . \end{cases} \tag{8}$$

This ensures that one of the summands in Equation (7) will always be equal to 0. $\tilde{\Pr}$ and α depend on the particular discounting algorithm and represent the adjusted probabilities and the normalised freed probability mass, respectively:

$$\alpha(h) = \begin{cases} 1 - \sum_{w_i} \tilde{\Pr}(w_i|h) & \text{if } \mathrm{C}(h) > 0 \\ 1 & \text{otherwise} \ . \end{cases} \tag{9}$$

The second case in Equation (9) covers events where the $(N-1)$-gram history is not available. The lower ordered distribution is used unweighted in such cases. Since lower ordered distributions are probabilistic by definition, the whole model keeps this property.

The back-off recursion is terminated either by the (undiscounted) unigram distribution

$$\hat{\Pr}(w_i) = \Pr(w_i) \ . \tag{10}$$

or by a uniform distribution which handles out-of-vocabulary items. Such a uniform distribution involves a non-probabilistic model, since any number of out-of-vocabulary items is possible:

$$\hat{\Pr}(\varepsilon) = \Pr_{unif}(\varepsilon) = \frac{1}{\sum_{b \in \Sigma} 1} \ . \tag{11}$$

Back-off smoothing is compatible with all discounting algorithms. We use Witten-Bell discounting as explained above.

3 Formal Preliminaries

In this section, we define the formal apparatus used in the remainder of this article. We start with the notion of a *semiring*, define weighted rational languages and transductions, move to the definition of weighted finite-state acceptors and transducers and a number of operations defined on them and finally clarify the relationship between weighted languages on the one and finite automata on the other hand.

3.1 Semirings

The weights of languages, transductions and automata are expressed in terms of a semiring. The advantage in doing so lies in the abstraction and well-definedness of operations and algorithms for different types of weights (e.g. [19, 25, 24]).

Definition 3 (Semiring)**.** *A structure* $\mathcal{K} = \langle \mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1} \rangle$ *is a semiring if*

1. $\langle \mathbb{K}, \oplus, \overline{0} \rangle$ *is a commutative monoid with* $\overline{0}$ *as the identity element for* $\oplus$*,*
2. $\langle \mathbb{K}, \otimes, \overline{1} \rangle$ *is a monoid with* $\overline{1}$ *as the identity element for* $\otimes$*,*
3. $\otimes$ *distributes over* $\oplus$ *(distribution of one operation over another will be denoted by* $\succ$*, e.g.* $\otimes \succ \oplus$*) , and*
4. $\overline{0}$ *is an annihilator for* $\otimes$*:* $\forall a \in \mathbb{K}, a \otimes \overline{0} = \overline{0} \otimes a = \overline{0}$ *.*

Examples for semirings are the *boolean* semiring $\mathcal{B} = \langle \{0,1\}, \vee, \wedge, 0, 1 \rangle$, the *real* semiring $\mathcal{R} = \langle \mathbb{R} \cup \{\infty\}, +, \cdot, 0, 1 \rangle$, the *log* semiring $\mathcal{L} = \langle \mathbb{R} \cup \{\infty\}, +_{\log}, +, \infty, 0 \rangle$[2] or the *tropical* semiring $\mathcal{T} = \langle \mathbb{R}^+ \cup \{\infty\}, \min, +, \infty, 0 \rangle$. A special significance in the remainder of this work lays on the *probability* semiring $\mathcal{P} = \langle \mathbb{R}^+ \cup \{\infty\}, +, \cdot, 0, 1 \rangle$ since its properties make it suitable for representing probabilities.[3]

To be well-defined, some operations on languages and automata demand particular properties of the used semirings. See [19] for a detailed summary on semirings and their properties. For the scope of this article, we need the definitions of *idempotency*, *divisibility*, *commutativity* and *completeness*.

Definition 4 (Idempotent Semiring)**.** *A semiring* $\mathcal{K}$ *is called idempotent if* $a \oplus a = a$ *for all* $a \in \mathbb{K}$*.*

Definition 4 means that in case of non-idempotent semirings the $\oplus$ operation is effectively additive in a sense that it sums weights. The probability and the log semiring are non-idempotent.

Definition 5 (Division Semiring)**.** *A semiring* $\mathcal{K}$ *is a division semiring iff* $\forall a \in \mathbb{K} \setminus \{\overline{0}\}$*,* $\exists! b \in \mathbb{K}$ *such that* $a \otimes b = \overline{1}$*.*

Divisibility (cf. [9]) is a formalisation of the demand for closure under *multiplicative inversion* needed for division of elements in $\mathbb{K}$. This property is adapted from a special class of *rings* called the *divisible rings*.

Definition 6 (Commutative Semiring)**.** *A semiring is said to be commutative when the* $\otimes$ *operation is commutative; that is* $\forall a, b \in \mathbb{K}, a \otimes b = b \otimes a$*.*

The requirement that sums of an infinite number of elements are well defined is expressed as *completeness* (e.g. [10]).

Definition 7 (Complete Semiring)**.** *A semiring* $\mathcal{K}$ *is called complete if it is possible to define sums for all families* $(a_i | i \in I)$ *of elements in* $\mathbb{K}$*, where* I *is an arbitrary index set, such that the following conditions are satisfied:*

[2] $a +_{\log} b =_{def} -\log(2^{-a} + 2^{-b})$

[3] The terms 'probability semiring' and 'real semiring' are interchanged freely in the corresponding literature. The following distinction seems sensible: Since real numbers can be both positive and negative, the real semiring should be defined over $\mathbb{R}$. Probability on the other hand will always be positive, thus in $\mathbb{R}^+$.

(i) $\bigoplus_{i\in\emptyset} a_i = \overline{0}$, $\bigoplus_{i\in\{j\}} a_i = a_j$, $\bigoplus_{i\in\{j,k\}} a_i = a_j \oplus a_k$ *for* $j \neq k$,

(ii) $\bigoplus_{j\in J}(\bigoplus_{i\in I_j} a_i) = \bigoplus_{i\in I} a_i$, *if* $\bigcup_{j\in J} I_j = I$ *and* $I_j \cap I_{j'} = \emptyset$ *for* $j \neq j'$,

(iii) $\bigoplus_{i\in I}(c \otimes a_i) = c \otimes (\bigoplus_{i\in I} a_i)$, $\bigoplus_{i\in I}(a_i \otimes c) = (\bigoplus_{i\in I} a_i) \otimes c$.

In the following, we restrict our attention to commutative, divisible, complete and non-idempotent semirings. For better transparency, we primarily use the real semiring $\mathcal{R}$ and the probability semiring $\mathcal{P}$ in definitions, lemmas and proofs, but also use the more general semiring notation with $\oplus$ and $\otimes$.[4]

We will denote semirings with capital letters in calligraphic character style like $\mathcal{P}$, $\mathcal{K}$.

3.2 Weighted Rational Languages and Transductions

Every formal language can be represented as a weighted language.

Definition 8 (Weighted Language)**.** *A weighted language* $\mathcal{L}$ *is a mapping* $\Sigma^* \to \mathcal{K}$, *where* Σ *denotes a finite set of symbols (called the alphabet) and* $\mathcal{K}$ *a semiring.*

This definition applies to all formal languages. The different types of languages are distinguished by the operations that are allowed to construct the subset of Σ^* from the singletons in Σ (see below).

Definition 9 (Weighted Transduction)**.** *A weighted transduction* $\mathcal{S}$ *is a mapping* $\Sigma^* \times \Gamma^* \to \mathcal{K}$, *where* Σ *and* Γ *denote finite sets of symbols (called the input and the output alphabet, resp.) and* $\mathcal{K}$ *a semiring.*

Weighted rational languages (WRL) and *weighted rational transductions* (WRT) are a proper subset of the weighted languages and transductions. They can be constructed from singletons in a finite alphabet Σ using *scaling*, *union*, *concatenation*, *composition* and *closure* [26]. In addition to these, we use a set of operations on WRLs and WRTs summarised in Table 1.

Definition 10 equates any WRL with its identity transduction.

Definition 10 (Identity Transduction)**.** *Given a WRL* $\mathcal{L} : \Sigma^* \to \mathcal{K}$, *its identity transduction* $ID(\mathcal{L}) : \Sigma^* \times \Sigma^* \to \mathcal{K}$ *is defined as:*

$$\forall x, y \in \Sigma^*,\ ID(\mathcal{L})(x,y) = \begin{cases} \mathcal{L}(x) & \text{if } x = y \\ \overline{0} & \text{otherwise} \end{cases} .$$

An often used complex operation is *application*:

Definition 11 (Application)**.** *The application of a WRT* $\mathcal{S} : \Sigma^* \times \Gamma^* \to \mathcal{K}$ *to a WRL* $\mathcal{L} : \Sigma^* \to \mathcal{K}$ *is a mapping* $\mathcal{S}[\mathcal{L}] : \Gamma^* \to \mathcal{K}$ *defined by*

$$\forall y \in \Gamma^*,\ \mathcal{S}[\mathcal{L}](y) = \bigoplus_{x\in\Sigma^*} \mathcal{L}(x) \otimes \mathcal{S}(x,y) .$$

[4] In practice, $\mathcal{P}$'s isomorphic counter part, the log semiring $\mathcal{L}$ would be used instead for reasons of numerical stability.

Table 1: Operations on WRLs and WRTs

Let $\mathcal{S}\colon \Sigma^* \times \Delta^* \to \mathcal{K}$, and $\mathcal{Q}\colon \Delta^* \times \Gamma^* \to \mathcal{K}$, denote two WRTs and let $\mathcal{L}_1\colon \Sigma^* \to \mathcal{K}$, and $\mathcal{L}_2\colon \Sigma^* \to \mathcal{K}$, denote two WRLs.[a] Let a, b and c, d be chosen from the same alphabet (augmented with ε), respectively. For $\mathcal{S}$ (also $\mathcal{S}_1, \mathcal{S}_2$), let the operands x and y range over Σ^* and Δ^*, resp. For $\mathcal{Q}$, let x and y range over Δ^* and Γ^*, resp. For $\mathcal{L}_1$ and $\mathcal{L}_2$, $x, y \in \Sigma^*$.

singleton	$\{(a,c)\}(b,d)$	$=$	$\overline{1}$ if $a = b$ and $c = d$, $\overline{0}$ otherwise
singleton	$\{a\}(b)$	$=$	$\overline{1}$ if $a = b$, $\overline{0}$ otherwise
union (sum)	$(\mathcal{S}_1 \cup \mathcal{S}_2)(x,y)$	$=$	$\mathcal{S}_1(x,y) \oplus \mathcal{S}_2(x,y)$
concatenation	$(\mathcal{S}_1 \cdot \mathcal{S}_2)(x,y)$	$=$	$\bigoplus_{tu=x, vw=y} \mathcal{S}_1(t,v) \otimes \mathcal{S}_2(u,w)$
scaling	$k\mathcal{Q}(x,y)$	$=$	$k \otimes \mathcal{Q}(x,y) \quad (k \in \mathcal{K})$
power	$\mathcal{Q}^0(\varepsilon,\varepsilon)$	$=$	$\overline{1}$
	$\mathcal{Q}^0(x \neq \varepsilon, y \neq \varepsilon)$	$=$	$\overline{0}$
	$\mathcal{Q}^{n+1}(x,y)$	$=$	$(\mathcal{Q} \cdot \mathcal{Q}^n)(x,y)$
closure	$\mathcal{Q}^*(x,y)$	$=$	$\bigoplus_{k \geq 0} \mathcal{Q}^k(x,y)$
composition	$(\mathcal{S} \circ \mathcal{Q})(x,y)$	$=$	$\bigoplus_{z \in \Delta^*} \mathcal{S}(x,z) \otimes \mathcal{Q}(z,y)$
1st *projection*	$\pi^1(\mathcal{S})(x)$	$=$	$\bigoplus_{y \in \Delta^*} \mathcal{S}(x,y)$
2nd *projection*	$\pi^2(\mathcal{S})(y)$	$=$	$\bigoplus_{x \in \Sigma^*} \mathcal{S}(x,y)$
crossproduct	$(\mathcal{L}_1 \times \mathcal{L}_2)(x,y)$	$=$	$\mathcal{L}_1(x) \otimes \mathcal{L}_2(y)$
intersection	$(\mathcal{L}_1 \cap \mathcal{L}_2)(x)$	$=$	$\mathcal{L}_1(x) \otimes \mathcal{L}_2(x)$

[a] Using the identity transduction from Definition 10, the operations union, concatenation, power, scaling, and closure also apply to weighted rational languages.

Application is a short-cut for composing the identity transduction of $\mathcal{L}$ with $\mathcal{S}$ and taking the 2nd projection afterwards.

Definition 12 (Language Projection)**.** *Given a WRL $\mathcal{L} : \Sigma^* \to \mathcal{K}$, the language projection of $\mathcal{L}$ – denoted by $\pi^L(\mathcal{L})$ – is defined as*

$$\forall x \in \Sigma^*,\ \pi^L(\mathcal{L})(x) = \begin{cases} \overline{1} & \text{if } \mathcal{L}(x) \neq \overline{0} \\ \overline{0} & \text{otherwise} \end{cases} .$$

Since language projection is an operation which only replaces the weights of its operand, WRLs and WRTs are closed under it (see the result for *length preserving* homomorphisms in [8]).

Definition 13 ($\otimes$-negation)**.** *Given a WRL $\mathcal{L} : \Sigma^* \rightarrow \mathcal{K}$ over a division semiring $\mathcal{K}$, the $\otimes$-negation of $\mathcal{L}$ – denoted by $\mathcal{L}^{-\bar{1}}$ – is defined as*

$$\forall x \in \Sigma^*,\ \mathcal{L}^{-\bar{1}}(x) = \begin{cases} a & \text{if } \mathcal{L}(x) \neq \bar{0} \ \wedge \ a \otimes \mathcal{L}(x) = \bar{1} \\ \bar{0} & \text{otherwise} \ . \end{cases}$$

Further on, we use capital script letters like $\mathcal{L}$, $\mathcal{P}$ to denote weighted languages and transductions.

3.3 Weighted Finite-State Automata

Every WRL and every WRT can be represented by at least one weighted finite-state acceptor or transducer, respectively.

Definition 14 (WFSA)**.** *A* weighted finite-state acceptor *(henceforth WFSA, cf. [24]) $\mathfrak{A} = \langle \Sigma, Q, q_0, F, E, \lambda, \rho \rangle$ over a semiring $\mathcal{K}$ is a 7-tuple with*

1. *Σ, the finite input alphabet,*
2. *Q, the finite set of states,*
3. *$q_0 \in Q$, the start state,*
4. *$F \subseteq Q$, the set of final states,*
5. *$E \subseteq Q \times Q \times (\Sigma \cup \{\varepsilon\}) \times \mathcal{K}$, the set of transitions,*
6. *$\lambda \in \mathcal{K}$, the initial weight, and*
7. *$\rho : F \rightarrow \mathcal{K}$, the final weight function mapping final states to elements in $\mathcal{K}$.*

An extension of WFSAs are the *weighted finite-state transducers.*

Definition 15 (WFST)**.** *A* weighted finite-state transducer *(henceforth WFST) $\langle \Sigma, \Delta, Q, q_0, F, E, \lambda, \rho \rangle$ over a semiring $\mathcal{K}$ is a 8-tuple with*

1. *Σ, Q, q_0, F, λ and ρ are defined in the same manner as in the case of WFSAs,*
2. *Δ, the finite output alphabet, and*
3. *$E \subseteq Q \times Q \times (\Sigma \cup \{\varepsilon\}) \times (\Delta \cup \{\varepsilon\}) \times \mathcal{K}$, the set of transitions.*

The weight assigned by a WFSA $\mathfrak{A}$ to a string $x \in \Sigma^*$ is determined by Definition 16.

Definition 16 (Weight of a String)**.** *Let $\mathfrak{A} = \langle \Sigma, Q, q_0, F, E, \lambda, \rho \rangle$ be a WFSA over a semiring $\mathcal{K}$. Let π be a path in $\mathfrak{A}$, that is, a sequence of adjacent transitions. Let $n[\pi]$ denote the state reached at the end of π. Let $\Pi(Q_1, x, Q_2)$ denote the set of all paths from $q_1 \in Q_1$ to $q_2 \in Q_2$ labeled with $x \in \Sigma^*$. Let $\omega(\pi)$ denote the $\otimes$-multiplication of the weights of the transitions along the path π. The weight assigned to a string $x \in \Sigma^*$ by $\mathfrak{A}$, denoted by $[\![x]\!]^{\mathfrak{A}}$, is defined as:*

$$[\![x]\!]^{\mathfrak{A}} = \bigoplus_{\pi \in \prod(\{q_0\}, x, F)} \lambda \otimes \omega(\pi) \otimes \rho(n(\pi)) \ .$$

A WFSA is called *unambiguous*, if there is for each input string x at most a single path in $\mathfrak{A}$. As a special case, each state q in a *deterministic* WFSA has at most a single target state for each $a \in \Sigma$. Note that in case of unambiguous/deterministic WFSAs, the $\oplus$-operation in Definition 16 has no effect, since there is for every input string only a single path from q_0 to a final state.

In addition to the automata-algebraic operations like union, intersection, concatenation etc., we use three equivalence operations, e.g. operations which only change the structure of a WFSA but not the weighted language it accepts, parametrised with respect to a semiring $\mathcal{K}$: *rm-*$\varepsilon_{\mathcal{K}}$ for ε-removal, $det_{\mathcal{K}}$ for determinisation of WFSAs, and $min_{\mathcal{K}}$ for minimisation. We omit the subscript for the semiring if it is understood from the context.

If $\mathcal{K}$ is a divisible semiring, we denote by $neg^{\otimes}_{\mathcal{K}}$ the operation, which replaces the initial weight λ and each transition and final state weight a of a WFSA $\mathfrak{A}$ by its multiplicative inverse, denoted by $\lambda^{-\bar{1}}$ and $a^{-\bar{1}}$ respectively. Note that $\mathfrak{A}$ must be at least unambiguous to obtain the correct result corresponding to Definition 13. Although not every WFSA can be determinised [21], those WFSAs to which we apply $neg^{\otimes}_{\mathcal{K}}$ have an equivalent deterministic counterpart.

Typographically, we will render acceptors and transducers with letters in Gothic type, for example $\mathfrak{C}$, $\mathfrak{K}$.

4 N-Gram Counting

As shown in Section 2, frequencies of events are necessary for creating N-gram word models. This section shows how to obtain these frequencies.

4.1 Text Corpora as Weighted Finite-State Automata

Text corpora can be easily represented as acyclic weighted finite state acceptors over the real semiring. This approach is advantageous since acyclic WFSAs always admit equivalence transformations like determinisation and minimisation [21]. Fig. 1 shows a WFSA $\mathfrak{K}$ constructed from a toy corpus.[5]

[5] We adopt the convention that transition labels are of the form a/w in case of acceptors and $a : b/w$ when depicting transducers: $a \in \Sigma \cup \{\varepsilon\}$ denotes the input symbol of the transition, $b \in \Delta \cup \{\varepsilon\}$ is its output symbol and $w \in K$ its weight. In the context of an WFST, a transition labeled with a stands for the identity transduction $a : a$. Similar, the final weight $\rho(p)$ assigned to

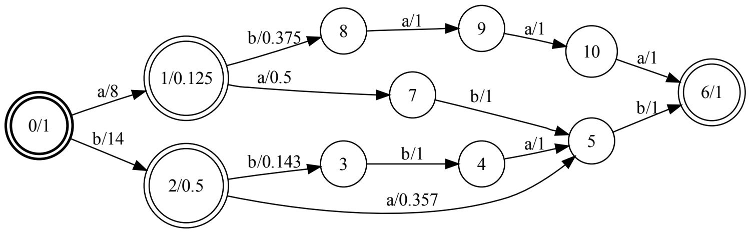

Figure 1: A toy corpus over $\Sigma = \{a, b\}$ represented as a WFSA $\mathfrak{K}$.

The number of occurrences of a given sentence s can be computed along Definition 16; for example $[\![aabb]\!]^{\mathfrak{K}} = 1 \cdot 8 \cdot 0.5 \cdot 1 \cdot 1 \cdot 1 = 4$.

4.2 N-gram Counting

An approach for counting N-grams with WFSTs has been proposed in [2]. We adopt this approach and repeat the resulting definitions using the notation introduced in Section 3. For the purpose of counting N-grams, a special transducer which realises a rational transduction $\mathcal{F} : \Sigma^* \times \Sigma^* \to \mathcal{R}$ is used:

$$\forall x, y \in \Sigma^*,\ \mathcal{F}(x, y) = ((\Sigma \times \{\varepsilon\})^* \cdot\ ID(\mathcal{L})\ \cdot (\Sigma \times \{\varepsilon\})^*)\ (x, y) \tag{12}$$

where $\mathcal{L}$ is a WRL mapping Σ^* to $\mathcal{R}$, such that the number of strings x with $\mathcal{L}(x) \neq 0$ is finite. In the case of N-gram counting, the domain of $\mathcal{L}$ needs to be Σ^N (in which case we write $\mathcal{F}_N(x, y)$). To gain some information about which words occurred at the beginning or end of a sentence in the corpus, we augment the alphabet Σ with two special symbols <s> and </s> marking the beginning and the end of each sentence, respectively. For that purpose, we prefix our corpus WRL with $N - 1$ <s>-symbols and append $N - 1$ </s>-symbols at its end (this also simplifies the computation of the conditional probabilities, see Section 6). Fig. 2 shows an example for $N = 3$. Note that the delimiter symbols are treated in an optimised manner.

Counting is performed by applying the counting WRT $\mathcal{F}_N$ to the weighted language $\mathcal{K}$ given by the corpus:

Definition 17 (N-gram counting)**.** *Given a WRL $\mathcal{K} : \Sigma^* \to \mathcal{R}$ representing a corpus, the N-gram counts $\mathcal{C}_N : \Sigma^* \to \mathcal{R}$ are obtained by:*

$$\mathcal{C}_\mathcal{N} = \mathcal{F}_N[\mathcal{K}]\ .$$

a final state p (printed as a double circle) is stated after /. If the weight is omitted, it is assumed to be $\bar{1}$.

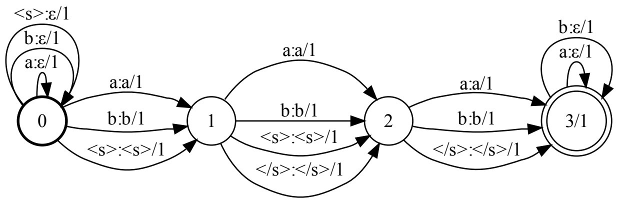

Figure 2: Transducer for counting trigrams over $\Sigma = \{a, b, \text{<s>}, \text{</s>}\}$.

We also call $\mathcal{C}_N$ an *N-gram count WRL*. For details on the procedure and a proof of its correctness we refer the reader to [2].

The trigram counts for the example corpus (Figure 1) are shown in Figure 3 (after optimising – that is removal of ε-transitions, determinisation, and minimisation – the corresponding WFSA). Note that for the purpose of demonstrating non-robust language models first (cf. Section 6) we have chosen a corpus over $\Sigma = \{a, b, <\text{s}>, <\text{/s}>\}$ which contains each meaningful trigram in Σ^N at least once resulting in an almost complete WSA.[6] Note that trigrams ending in <s> or starting with </s> cannot exist.

To get the count $\mathrm{C}(w_1 \ldots w_N)$ associated with a specific N-gram $w_1 \ldots w_N$ we compute $\llbracket w_1 \ldots w_N \rrbracket^{\mathfrak{C}_N}$ – the weight assigned to $w_1 \ldots w_N$ by $\mathfrak{C}_N$ according to Definition 16. For example, $\llbracket ab\text{</s>} \rrbracket$ of Figure 3 is $1 \cdot 28 \cdot 0.5 \cdot 0.5 \cdot 1 = 7$.

4.3 Implementation and Complexity

The structure and therefore the size of the WFST $\mathfrak{F}_N$ corresponding to $\mathcal{F}_N$ depends on the model parameter N and the size of the underlying alphabet. Its state number $|Q|$ equals $N + 1$ and the number of transitions $|E|$ is $|\Sigma|(N + 2)$. Its space complexity is within $\mathrm{O}(N|\Sigma|)$, thus the size of $\mathfrak{F}_N$ may become problematic for huge alphabets. As already suggested in [2], a solution to this problem are *lazy* automata, the states and transitions of which are constructed on-demand. Such automata are usually obtained from lazy versions of the finite-state algorithms. For example, an algorithm for the lazy composition of WRTs is presented in [28]. The drawback of such approaches is that the basic operands have to be explicitly represented.

Other approaches (among others, see [4]) try to construct automata *virtually* right from the beginning. Regularities in their structure are used to define states

[6] A (W)FSA is called *complete* with respect to an alphabet Σ if each state has outgoing transitions for each symbol $a \in \Sigma$.

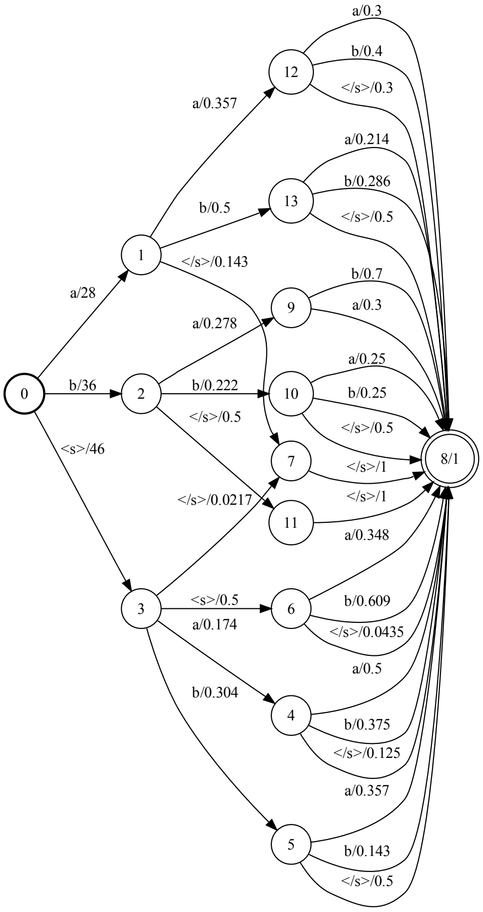

Figure 3: Trigrams in the toy corpus after optimisation.

and transitions implicitly by some calculation specification.

The simple structure of $\mathfrak{F}_N$ makes it suitable for a virtual construction: The set of states Q is simply $\bigcup_{q=0}^{N}\{q\}$ with N being the only final state. The set of transitions E has three different subsets: E_i, containing all transitions from the initial state, E_m, containing all transitions from non-initial and non-final states and E_f containing all transitions to the final state. Transitions in E_m for example lead from state q to state $q+1$ with each symbol $a \in \Sigma$ while emitting this symbol. The formal construction of $\mathfrak{F}_N$ can be found in Definition 35 in Appendix B.

Definition 35 enables a virtual construction. Implementations of access functions to states and transitions work in O(1) time while consuming only a constant amount of memory. We have implemented this special representation of $\mathfrak{F}_N$ within the framework of [12].

Given a corpus WFSA $\mathfrak{K}$ and an N-gram counter $\mathfrak{F}_N$, counting is performed most efficiently by the following sequence of automata operations:

$$\mathfrak{C}_N = min(det(rm\text{-}\varepsilon(\pi^2(\mathfrak{K} \circ \mathfrak{F}_N)))) \ . \tag{13}$$

Since the number of N-gram paths after composition is bounded by $|\mathfrak{K}|$ and since the result is acyclic, ε-removal, determinisation (which is essentially the construction of a trie from the found N-grams), and minimisation (including weight-pushing) can be performed in O($|\mathfrak{K}|$) time [27, 25, 24, 13].[7]

5 Probabilisation

The next step in constructing an N-gram language model is to compute the conditional probabilities of the events according to their frequency. This is done by normalising their counts (this equation is also called *maximum likelihood estimation*, see [16]):

$$\Pr(w_i | w_{i-N+1}^{i-1}) = \frac{\mathrm{C}(w_{i-N+1}^{i-1} \cdot w_i)}{\sum\limits_{a \in \Sigma} \mathrm{C}(w_{i-N+1}^{i-1} \cdot a)} \ . \tag{14}$$

Thus, the frequency of an N-gram is divided by the sum of the frequencies of all N-grams sharing the same $(N-1)$-gram prefix.

5.1 Conditional Probabilities

In order to normalise the N-gram counts as stated in equation (14), the weights of all N-grams sharing the same $(N-1)$-gram prefix have to be collected. Both parts of the division need to have the same language projection to guarantee that no N-grams are lost. The N-grams are therefore 'reweighted' by their corresponding collected prefix weights. This reweighting is done by a suffix expansion performed by a WRT $\mathcal{E}_N^k : \Sigma^N \times \Sigma^N \to \mathcal{R}$ which maps all N-gram suffixes of length k to each other, what effectively assigns each weight to every symbol.

Definition 18 (Suffix expansion)**.** *Given a finite alphabet* Σ *and model parameters* $N > 0$ *and* $k \leq N$*, a WRT* $\mathcal{E}_N^k : \Sigma^N \times \Sigma^N \to \mathcal{R}$ *is defined as*

$$\forall x, y \in \Sigma^N, \ \mathcal{E}_N^k(x, y) = (ID(\Sigma^{N-k}) \cdot (\Sigma \times \Sigma)^k)\ (x, y) \ .$$

[7] $|\mathfrak{A}| = |Q_{\mathfrak{A}}| + |E_{\mathfrak{A}}|$, that is, the size of a WFSA $\mathfrak{A}$ is measured in terms of the size of its state set and its number of transitions.

The first $N-k$ steps of this transduction are an identity mapping. The following k steps create – to speak in terms of the underlying WFST – non-deterministic paths: The crossproduct of Σ with Σ results in $|\Sigma|$ transitions for every symbol $a \in \Sigma$. The corresponding transducer for $\mathcal{E}_3^1$ is shown in Figure 4.[8] By applying $\mathcal{E}_N^1$ to the N-gram counts, the weights of all N-grams are expanded. The chosen $k = 1$ cares for the summing over the unigram suffixes and the N-grams bear the sum of the weight of the N-grams sharing the same $(N-1)$-gram prefixes as demanded by Equation (14). The extended weights are $\otimes$-negated and intersected with the N-gram counts to perform the normalisation. Given the N-gram counts $\mathcal{C}_\mathcal{N}$ as computed in Section 4, $\mathcal{P}^c_N(\mathcal{C}_\mathcal{N}) : \Sigma^N \to \mathcal{R}, w = w_1^N \mapsto \Pr(w_N|w_1^{N-1})$ implements this series of rational operations.

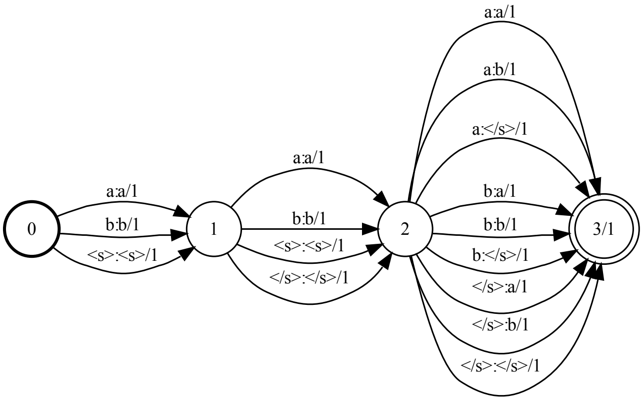

Figure 4: The unigram suffix expansion for trigrams $\mathfrak{E}_3^1$ for $\Sigma = \{a, b, \text{<s>}, \text{</s>}\}$.

Definition 19 (Conditional N-gram probabilisation). *Given a WRL* $\mathcal{C}_N : \Sigma^N \to \mathcal{R}, \quad w_1^N \mapsto \mathrm{C}(w)$, $\mathcal{P}^c_N(\mathcal{C}_N)$ *is defined as*[9]

$$\mathcal{P}^c_N(\mathcal{C}_N) = \left(\mathcal{C}_N \cap (\mathcal{E}^1_N[\mathcal{C}_N])^{-\bar{1}}\right) .$$

An example of the application of Definition 19 is shown in Figure 5.

In Figure 5, the probability of seeing a b after having seen an ab – that is, $\Pr(b|ab) = [\![abb]\!]$ – is 0.4.

[8] Again, some transitions related to the delimiters were removed for reasons of clarity.

[9] Note that the *joint N-gram probabilisation* (which reflects the joint probability of each N-gram), is computed by $\mathcal{P}^j_N(\mathcal{C}_N) = \left(\mathcal{C}_N \cap (\mathcal{E}^N_N[\mathcal{C}_N])^{-\bar{1}}\right)$. The language weight of such an probabilisation, that is $\bigoplus_{x \in \mathcal{C}_N} \mathcal{P}^j_N(\mathcal{C}_N)(x)$, equals $\bar{1}$.

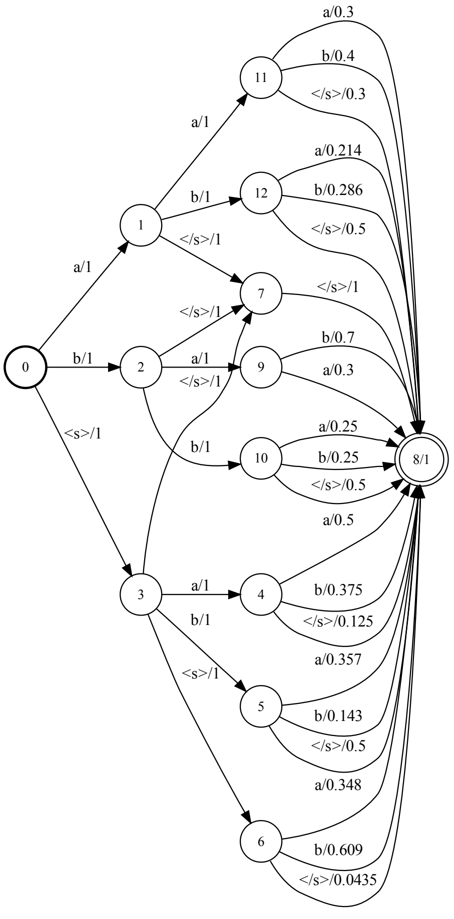

Figure 5: Conditional probabilised trigrams from the example corpus.

Lemma 1 (Correctness of conditional N-gram probabilisation)**.** *Definition 19 computes the conditional probability of each N-gram as a special case of Equation (14) (with $i = N$):*

$$\Pr(w_N|w_1^{N-1}) = \frac{\mathrm{C}(w_1^N)}{\sum\limits_{a\in\Sigma} \mathrm{C}(w_1^{N-1} \cdot a)} \ . \tag{15}$$

Proof. See Appendix A. □

Note that an advantage of the automata/language theoretic approach is that Definition 19 computes the conditional probabilities of all found N-grams simultaneously.

5.1.1 Implementation and Complexity

The most efficient implementation of Definition 19 in terms of WFSA operations is the following:

$$\mathfrak{C}_N \cap neg^{\otimes}(min(det(\pi^2(\mathfrak{C}_N \circ \mathfrak{E}_N^1)))) \ . \tag{16}$$

A problem could arise through the constant factor associated with the alphabet size in Definition 18, since the number of transitions in a WFSA corresponding to $(\Sigma\times\Sigma)^k$ is $|\Sigma|^{2k}$. So the approach may become unfeasible in case of the big alphabet sizes commonly encountered in corpus linguistics. The composition operation $\circ$ maps every transition t in $\mathfrak{C}_N$ leading to a final state to $|\Sigma|$ transitions in the result. Since the operand of $neg^{\otimes}$ must be deterministic, all transitions resulting from suffix expansion must be (additively) combined by determinisation.

To get rid of the constant introduced by the size of the alphabet, we define a special symbol <?>, called the *default symbol* (see [5]). During intersection and composition, <?> matches every unmatched symbol labeling a transition leaving a state q. The definition of suffix expansion is then changed to the one in Definition 20:

Definition 20 (Revised suffix expansion). *Given two finite alphabets Σ and Δ and model parameters $N > 0$ and $k \le N$, a WRT $\mathcal{E}_N^{k,\Delta} : \Sigma^N \times (\Sigma^{N-k} \cdot \Delta^k) \rightarrow \mathcal{R}$ is defined as*

$$\forall x, y \in \Sigma^N,\ \mathcal{E}_N^{k,\Delta}(x,y) = (ID(\Sigma^{N-k}) \cdot (\Sigma \times \Delta)^k)\ (x,y) \ .$$

Note that $\mathcal{E}_N^k$ is a special case of Definition 20. The special suffix expansion using <?> is then $\mathcal{E}_N^{k,\{<?>\}}$.

To reflect the special semantics of <? >, the implementations of $\cap$ and $\circ$ are changed to $\cap^{<?>}$ and $\circ^{<?>}$, respectively. Equation (16) becomes

$$\mathfrak{C}_N \cap^{<?>} neg^{\otimes}(min(det(\pi^2(\mathfrak{C}_N \circ^{<?>} \mathfrak{E}_N^1)))) \ . \tag{17}$$

The complexity of the suffix expansion, projection, determinisation and minimisation is then in $\mathrm{O}(|\mathfrak{C}_N|)$. If we assume that $\mathfrak{C}_N$ is deterministic, the complexity of the final intersection step is also in $\mathrm{O}(|\mathfrak{C}_N|)$, since both operands contain exactly the same N-grams (they have the same language projection), thus are isomorphic.

The possible types of symbols in a (W)FSA may be cross-classified according to Table 2. Following Table 2, the default symbol <? > can be seen as a conditionally interpreted input consuming symbol. We will need its non-consuming counterpart, the *failure transition symbol* ϕ (see [1]) in Section 7 to create robust back-off language models.

	+consuming	−consuming
+conditional	<?>	ϕ
−conditional	$a \in \Sigma$	ε

Table 2: A cross-classification of symbols labeling transitions in an FSA

In parallel to the counting WRT, it is possible to define a calculation for $\mathfrak{C}_N^{k,\Delta}$ which enables its virtual construction. The calculation is given in Definition 36 (see Appendix B).

We move to the creation of non-robust language models.

6 Creating Non-Robust Language Models

The result of the counting and the normalisation procedure $\mathcal{P}^c_N$ is a weighted language $\Sigma^N \rightarrow \mathcal{R}$. It assigns the conditional probability $\Pr(w_i|w_{i-N+1}^{i-1})$ to every N-gram in the corpus. A maximum likelihood model is characterised by the following equation:

$$\Pr(w_1^m) = \prod_{i=1}^{m} \Pr(w_i|w_{i-N+1}^{i-1}) \ . \tag{18}$$

It is a weighted language $\Sigma^* \rightarrow \mathcal{R}$. Therefore, $\mathcal{P}^c_N$ has to be transformed to accept sequences of any length. Simply taking its closure is not sufficient, since the result would be a mapping from $(\Sigma^N)^* \rightarrow \mathcal{R}$: every N-gram could be followed by any other N-gram, every input symbol would have to be processed N times (as illustrated in example 1) and only strings with a length equal to a multiple of N would be in its domain.

Example 1 (Illustration of the necessary bigram overlapping)**.**

Given input		a		b		c
		w_1		w_2		w_3
$\Pr(w_1^3)$	$=$	$\Pr(a)$	$\cdot$	$\Pr(b\|a)$	$\cdot$	$\Pr(c\|b)$
To process (overlap)		a		ab		bc

To correctly reflect Equation (18), N-grams need to be overlapped in a way such that every $(N-1)$-gram suffix is simultaneously treated as an $(N-1)$-gram prefix. In order to achieve this, a specialisation of the concatenation operation called *overlapping* or *domino concatenation* is introduced.

Definition 21 (Domino (Overlapping) Concatenation)**.** *The overlapping concatenation of two WRTs $\mathcal{S} : \Sigma^* \times \Delta^* \rightarrow \mathcal{R}$ and $\mathcal{Q} : \Sigma^* \times \Delta^* \rightarrow \mathcal{R}$ – denoted by $\mathcal{S} \cdot_N \mathcal{Q}$ – is a mapping $\Sigma^* \times \Delta^* \rightarrow \mathcal{R}$ defined by*

$$\forall x \in \Sigma^*, \forall y \in \Delta^*,\ (\mathcal{S} \cdot_N \mathcal{Q})(x, y) = \bigoplus_{x=u\cdot v_1^{N-1}\cdot w, y=st} \mathcal{S}(u \cdot v_1^{N-1}, s) \otimes \mathcal{Q}(v_1^{N-1} \cdot w, t) \ .$$

The $\cdot_N$ operator is rational, as long as N is a constant.

6.1 N-Gram Models as WRLs

With the overlapping concatenation at hand, it is possible to use the closure of the conditional probability distribution as the basis of the N-gram model. It is used to filter non-overlapping sequences of N-grams. A transduction $\mathcal{D}_N : (\Sigma^N)^* \times \Sigma^* \to \mathcal{R}$ is defined which uses $\cdot_N$ in this way. To avoid their multiple processing, $\mathcal{D}_N$ deletes overlapping prefixes by simply omitting them from its output.

Definition 22 (N-gram Concatenator)**.** *Given a finite alphabet Σ, the N-gram concatenator is a WRT $\mathcal{D}_N : (\Sigma^N)^* \times \Sigma^* \to \mathcal{R}$, defined as*

$$\forall x, y \in \Sigma^*,\ \mathcal{D}_N(x,y) \ = \left(\Sigma^N \cup \left(\Sigma^N \ \cdot_N \ \mathop{\bigodot\nolimits_N}_{i=0}^{\infty} \left(\bigcup_{w_1^N \in \Sigma^N} \{(w_1^N, w_N)\}\right)\right)\right) (x, y)\ .$$

Fig. 6 shows a trigram concatenator for $\Sigma = \{a, b\}$. Note that the N-gram concatenator factors out the structure of an N-gram model (cf. [14], p.83) and makes it available to the algebraic formalisation independently from the corpus under consideration.

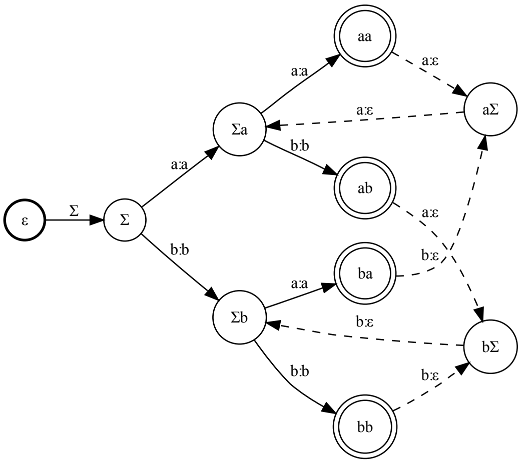

Figure 6: Trigram concatenator for $\Sigma = \{a, b\}$. States are labeled with their histories. The dashed transitions correspond to the overlaps.

To handle the special cases for $1 \leq M < N$ in Equation (18) uniformly, we prefix our input sentence with $N - 1$ <s>-symbols marking the sentence begin. Additionally, we postfix it with the same number of </s>-symbols marking its end, in order to guarantee that our language model seen as a WFSA has a unique final state (which is reached after reading the last </s>-symbol). For the model's structure, this means that only those N-grams starting with $(\text{<s>})^{N-1}$ and those ending in $(\text{</s>})^{N-1}$ may be accepted in the beginning and at the end, respectively. To reflect this, we *unfold* the closure of the conditional probabilities $\mathcal{P}^{c}_{N}$ by intersecting it with the WRL $\mathcal{U}_N$.

Definition 23 (Unfolding N-grams)**.** *Let* Σ *be an alphabet and* N *the model parameter.* $\mathcal{U}_N : \Sigma^* \to \mathcal{R}$ *is defined as:*

$$\forall x \in (\Sigma^N)^*,\ \mathcal{U}_N(x) = \left(\{\text{<s>}^{N-1}\} \cdot \Sigma \cdot (\Sigma^N)^* \cdot \Sigma \cdot \{\text{</s>}^{N-1}\}\right)(x)\ .$$

Definition 24 applies the N-gram concatenator $\mathcal{D}_N$ to the intersection of the closure of the probabilised N-grams and the unfolding WRL.

Definition 24 (Non-robust language models)**.** *Let* $\mathcal{C}_N$ *be an* N*-gram count WRL as defined in Definition 17, such that* $\mathcal{C}_N(x) \neq 0$*,* $\forall x \in \Sigma^N$*. The* non-robust language model $\mathcal{M}_N(\mathcal{C}_N)$ *is a weighted rational transduction* $\Sigma^* \to \mathcal{P}, x \in \Sigma^+ \mapsto \Pr(x)$

$$\mathcal{M}_N(\mathcal{C}_N) = \mathcal{D}_N[(\mathcal{P}^{c}_{N}(\mathcal{C}_N))^* \cap \mathcal{U}_N]\ .$$

Note that for the following theorem, we make the assumption that our input corpora are complete, that is, they contain every possible N-gram $w \in \Sigma^N$. We will relax this condition in Section 7.

Theorem 1 (Adequacy of Definition 24)**.** $\mathcal{M}_N(\mathcal{C}_N)(w)$ *correctly computes the decomposed conditional probability of Equation (18) for each delimited input string* w*.*

Proof. The proof is a special case (the two cases 1a) of the proof of Theorem 2 (cf. Appendix A). □

There is a relation between automata representing N-gram models and *de Bruijn graphs* [7]: A de Bruijn graph is a directed graph which represents the overlaps of sequences of a certain length n given a finite alphabet Σ. Each length n sequence of symbols in Σ is represented as a vertex in the graph. Let q denote the vertex for a sequence w_i^{i+n-1}, then q has a single edge for each symbol $a \in \Sigma$ connecting it to the vertex r representing $w_{i+1}^{i+n-1} \cdot a$. Thus, the structure of de Bruijn graphs is comparable to that of N-gram models over complete corpora.

6.2 Implementation and Complexity

Again, combining the WFSA for $\mathcal{P}^{c}_{N}$ and the WFST for $\mathcal{D}_N$ is basically application followed by optimisation:

$$\mathfrak{M}_N = \textit{rm-}\varepsilon\big(\pi^2\big(((\mathfrak{P}^{c}_{N})^* \cap \mathfrak{U}_N) \circ \mathfrak{D}_N\big)\big)\ . \tag{19}$$

If $(\mathfrak{P}^{c}_{N})^* \cap \mathfrak{U}_N$ is deterministic and since $\mathfrak{D}_N$ is input deterministic by definition, their composition will be input deterministic too. After taking the 2^{nd} projection, the ε-transitions resulting from the overlaps have to be removed. Although the result is a cyclic WFSA, its (unconnected) ε-subgraph will be acyclic, in fact, we will find a number of unconnected ε-chains of length $N-1$. These ε-transitions can be removed in linear time with respect to the size of the result of the application [23], which in turn is bounded by the size of $\mathfrak{P}^{c}_{N}$. Thus, the time complexity of Equation (19) is in $\mathrm{O}(|\mathfrak{P}^{c}_{N}|)$.

Even though $\mathfrak{D}_N$ depends only on the (constant) alphabet Σ and the model constant N, its space complexity is in $\mathrm{O}(\Sigma^{N-1})$, since $\mathfrak{D}_N$ has to keep track of the different histories of length $N-1$ to ensure correct overlaps. So, a naive implementation runs into difficulties even with moderate alphabet sizes. But we can do better if we exploit the regular structure of $\mathfrak{D}_N$ and replace actual states and transitions by functions computing them on demand. The trigram concatenator of Figure 6 is shown slightly modified in Figure 7. Labels of states have been replaced by state numbers and two additional states are introduced to simplify the virtual construction. In addition, we assume a bijective function $\mathtt{idx} : \Sigma \to \mathbb{N}$ mapping each alphabet symbol to a unique index r, $0 \leq r < |\Sigma|$. The labels of the transitions are replaced by their corresponding indices. Ignoring state 0, the first part of the

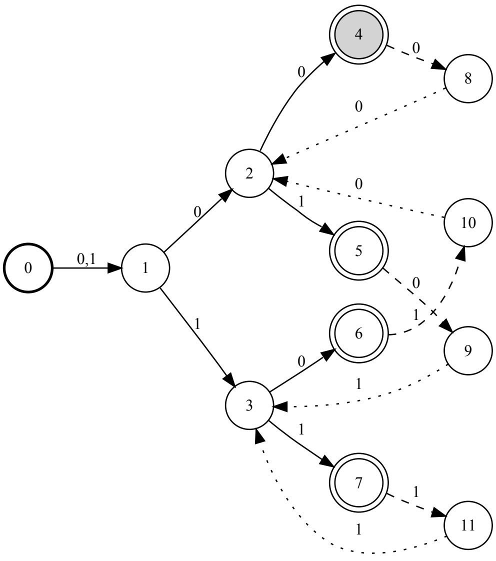

Figure 7: Trigram concatenator for $\Sigma = \{a, b\}$. States are labeled with numbers.

automaton shown in Figure 7 can be seen as a binary tree with root 1, yield $4 \dots 7$ and a consecutive labeling. The successor of a state q given an alphabet symbol a

can be calculated by $q * |\Sigma| + \texttt{idx}(a) - (|\Sigma| - 2)$ in the general case.

Example 2. Consider state 3 and symbol b with $\texttt{idx}(\text{b}) = 1$ in Figure 7. The correct destination state of the transition is state 7. Thus,

$$7 = 3 * 2 + 1 - (2 - 2).$$

The transitions within the tree part are denoted by E_t.

Transitions from states greater or equal than the first state of the yield q_y (state 4 in Figure 7) perform the overlap.

Definition 25 (Calculation of q_y)**.** *Given a finite alphabet Σ and a model parameter N, the state q_y is calculated as follows:*

$$q_y = \frac{|\Sigma|^{N-1} + (|\Sigma| - 2)}{|\Sigma| - 1} \ .$$

q_y is used to identify the states which do not allow branching. The transitions leaving those states are divided into the overlap transitions E_o and the loop transitions E_l. The computation of their destinations is simple, but one has to take care of the fact that only one symbol may be processed.
The complete calculation specification which enables a virtual construction of $\mathfrak{D}_N$ is given in Definition 37 in Appendix B. The virtual construction of $\mathfrak{U}_N$ is straightforward.
The next section focuses on *robust language models.*

7 Robust Language Models

Up to this point, the achieved models are only robust when based on corpora containing all possible N-grams which is an unrealistic assumption. As described in Section 2.2, smoothing methods have to be applied in order to solve this problem. Back-off smoothing can be described as 'relying on the highest order distribution which is available'. The following figure illustrates this behavior on the automata level (taken from [2]):

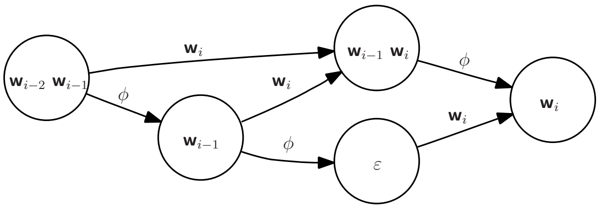

Figure 8: A trigram back-off model represented as a schematic FSA.

As suggested in [2], in those cases where – given a specific history – no transition for the next word w_i is available, a *failure transition* (marked by ϕ) to the nearest shorter history is traversed and w_i is processed if possible. If not, the history is shortened again until the history-less state is reached. Language models as achieved in Section 6 have to be extended to include such failure transitions to the lower ordered distributions. Following Section 2.2, it is necessary to apply a discounting algorithm in order to free probability mass for them.

For computing the probability of a (delimited) string w_1^m in a back-off model, we use the Markov probability decomposition as in Equation (18), but replace Pr with the back-off probability $\hat{\text{Pr}}$:

$$\Pr(w_1^m) = \prod_{i=1}^{m} \hat{\Pr}(w_i | w_{i-N+1}^{i-1}) \ . \tag{20}$$

7.1 Discounting

From the many existing discounting approaches, it is especially Witten-Bell discounting which is suited for modifying N-gram counts in a finite-state algebraic manner. The calculations for the discounted frequencies as well as for the freed frequency mass were given above in equations (5) and (6).

As explained above, Witten-Bell discounting uses the number of observed types following a history to estimate the probability of previously unseen events. Frequencies are discounted in relation to this number. Given a representation of N-gram counts, the number of types for each history can be computed with the help of the language projection (Definition 12) and the suffix expansion operator $\mathcal{E}_N^k$ (Definition 18). The idea is to first map all N-gram counts to 1 and then sum over the 1-gram suffixes.

Definition 26 (Witten-Bell Type Number)**.** *Given a WRL* $\mathcal{L} : \Sigma^N \to \mathcal{R}$*, a WRL* $\mathcal{T}_N : \Sigma^N \to \mathcal{R}$ *is defined as follows:*

$$\mathcal{T}_N(\mathcal{L}) = \mathcal{E}_N^1[\pi^L(\mathcal{L})] \ .$$

$\mathcal{T}_N$ directly corresponds to function T from Definition (1).

Lemma 2 (Correspondence of T and $\mathcal{T}_N$)**.** *Given a WRL* $\mathcal{L} : \Sigma^N \to \mathcal{R}$*,* $\forall w_1^N \in \Sigma^N : \mathcal{T}_N(\mathcal{L})(w_1^N) = \mathrm{T}(w_1^N)$*.*

Proof. See Appendix A. □

Definition 27 defines the analogon to N of Definition 2.

Definition 27 (Witten-Bell Token Number)**.** *Given a WRL* $\mathcal{L} : \Sigma^N \to \mathcal{R}$*, a WRL* $\mathcal{N}_N : \Sigma^N \to \mathcal{R}$ *is defined as follows:*

$$\mathcal{N}_N(\mathcal{L}) = \mathcal{E}_N^1[\mathcal{L}] \ .$$

Lemma 3 (Correspondence of N and $\mathcal{N}_N$)**.** *Given a WRL* $\mathcal{L} : \Sigma^N \to \mathcal{R}$*,* $\forall w_1^N \in \Sigma^N : \mathcal{N}_N(\mathcal{L})(w_1^N) = \mathrm{N}(w_1^N)$*.*

Proof. The proof is analogous to the proof of Lemma 2. □

The nominator of Equation (5) (which is at the same time the first summand of the denominator) has been used for obtaining conditional probabilities before (Section 5). Thus, everything needed for Witten-Bell discounting is at hand: we reconstruct Equation (5) using corresponding operations on WRLs. To reflect the N-gram discounting process, we actually operate on $\mathcal{C}_N$.

Definition 28 (Witten-Bell Discounting). *Given a WRL* $\mathcal{L} : \Sigma^N \to \mathcal{R}$*, we define* $\mathcal{W}_N^D(\mathcal{L}) : \Sigma^N \to \mathcal{R}, w \in \Sigma^N \mapsto \tilde{\mathrm{C}}(w)$ *as*

$$\mathcal{W}_N^D(\mathcal{L}) = \mathcal{L} \cap (\mathcal{N}_N(\mathcal{L}) \cap (\mathcal{N}_N(\mathcal{L}) \cup \mathcal{T}_N(\mathcal{L}))^{-\bar{1}}) \ ,$$

and $\mathcal{W}_N^R(\mathcal{L}) : \Sigma^N \to \mathcal{R}, w \in \Sigma^N \mapsto \mathrm{C}(w) - \tilde{\mathrm{C}}(w)$ *as*

$$\mathcal{W}_N^R(\mathcal{L}) = \mathcal{L} \cap (\mathcal{T}_N(\mathcal{L}) \cap (\mathcal{N}_N(\mathcal{L}) \cup \mathcal{T}_N(\mathcal{L}))^{-\bar{1}}) \ .$$

The second part of Definition 28 computes the freed frequency mass by reformulating Equation (6).

Again, we make use of the fact that the real semiring $\mathcal{R}$ is closed under multiplicative inverses to show that Definition 28 corresponds to the Witten-Bell discounted frequencies (resp. the freed frequency mass).

Lemma 4 (Reconstruction of Witten-Bell Discounting). *Given an N-gram count WRL* $\mathcal{C}_N : \Sigma^N \to \mathcal{R}$, $w_1^N \mapsto \mathrm{C}(w_1^N)$, $\mathcal{W}_N^D(\mathcal{C}_N)(w_1^N)$ *maps an N-gram to its Witten-Bell discounted frequency* $\tilde{\mathrm{C}}(w_1^N)$*.*

Proof. See Appendix A. □

The following equivalence holds:

Lemma 5 (Witten-Bell Decomposition). *Given an N-gram count WRL* $\mathcal{L} : \Sigma^N \to \mathcal{R}$, $\mathcal{W}_N^D(\mathcal{L}) \cup \mathcal{W}_N^R(\mathcal{L}) = \mathcal{L}$*.*

Proof. See Appendix A. □

An example of the discounting process is shown in Figure 9. Both parts of the Witten-Bell decomposition are used for reconstructing the back-off strategy as explained in the next section.

7.2 Back-off

The previously reserved frequency mass now has to be reallocated to the lower ordered distributions which need to be discounted as well (except the unigram distribution terminating the recursion). All involved distributions are then combined in a special representation to which the *robust overlapping concatenation* operator is applied.

The first step is to transform the adjusted frequencies into conditional probabilities. In principle, the procedure from Section 5 can be used with the difference that

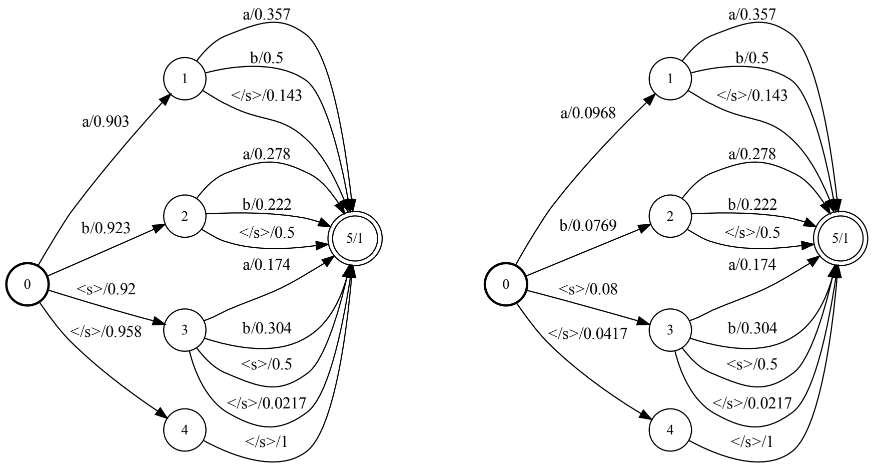

Figure 9: Witten-Bell decomposition for the bigrams of the corpus. The WFSA on the left is the discounted WFSA. Both WFSAs are already probabilised after Definition 29.

both have to be normalised in relation to the original counts instead of normalising them in relation to themselves. $\mathcal{P}^c_N$ is therefore modified to use the discounted frequencies (resp. the discounts, indicated by a second superscript) as the first argument of the integrated intersection operation.

Definition 29 (Witten-Bell Discounted Probabilities)**.** *Let $\mathcal{L}$ denote an N-gram count WRL $\Sigma^N \to \mathcal{R}$, then $\mathcal{P}^{c,D}_N : \Sigma^N \to \mathcal{R}$ is defined as*

$$\mathcal{P}^{c,D}_N(\mathcal{L}) = \mathcal{W}^D_N(\mathcal{L}) \cap (\mathcal{N}_N(\mathcal{L}))^{-\bar{1}},$$

and $\mathcal{P}^{c,R}_N : \Sigma^N \to \mathcal{R}$ is defined as

$$\mathcal{P}^{c,R}_N(\mathcal{L}) = \mathcal{W}^R_N(\mathcal{L}) \cap (\mathcal{N}_N(\mathcal{L}))^{-\bar{1}}.$$

$\mathcal{P}^{c,D}_N$ and $\mathcal{P}^{c,R}_N$ denote the Witten-Bell discounted probabilities and the freed probability mass of the N-grams when applied to $\mathcal{C}_\mathcal{N}$, respectively. Note that the union of $\mathcal{P}^{c,D}_N$ and $\mathcal{P}^{c,R}_N$ yields $\mathcal{P}^c_N$.

Lemma 6 (Witten-Bell Discounted Probabilities)**.** *Given $\mathcal{C}_N : \Sigma^N \to \mathcal{R}, w = w_1^N \in \Sigma^N \mapsto \mathrm{C}(w)$, $\mathcal{P}^{c,D}_N(\mathcal{C}_N)(w)$ and $\mathcal{P}^{c,R}_N(\mathcal{C}_N)(w)$ compute $\tilde{\Pr}(w_N|w_1^{N-1})$ and $\breve{\Pr}(w_N|w_1^{N-1})$, the Witten-Bell discounted probabilities and the freed probability mass, respectively.*

Proof. Lemma 6 results from Lemma 1 and Lemma 4. □

Lemma 7 (Union of $\mathcal{P}^{c,D}_N$ and $\mathcal{P}^{c,R}_N$)**.** *Let $\mathcal{L}$ denote an N-gram count WRL $\Sigma^N \to \mathcal{R}$:*

$$\mathcal{P}^{c,D}_N(\mathcal{L}) \cup \mathcal{P}^{c,R}_N(\mathcal{L}) = \mathcal{P}^c_N(\mathcal{L}) \ .$$

Proof. See Appendix A. □

7.2.1 The Unified Distribution

To create a model which contains all $N \ldots 1$-gram distributions, these have to be combined in some way. The aim is to enable the application of an overlapping filter - as in the non-back-off case - to the closure of the combination $\mathcal{Y}_N$ which therefore must, according to Equation (7), meet some requirements:

1. The single distributions must be discriminated from each other, since exactly one may account for a single event.
2. The single distributions must be ordered in a way that the back-off strategy is reflected.
3. The discounting factors $\alpha()$ of Equation (7) are context-dependent. They have to be assigned correctly.

The first point is realised by prefixing each M-gram distribution with $N - M$ α-symbols. Hence, their difference and hierarchy originates in the number of αs preceding them. α is a special symbol which is not part of Σ. It has no special semantics, is processed as any other symbol and will be deleted later. To comply with the third point, an α is appended to every $(M-1)$-gram prefix $(1 < M \leq N)$. This α will be identified with the back-off weight of the prefix it is attached to. We define the unified distribution $\mathcal{Y}_N$.

Definition 30 (Unified Distribution $\mathcal{Y}_N$)**.** *Given a WRL $\mathcal{L} : \Sigma^* \to \mathcal{R}$ representing a corpus, the combined representation of all $1 \ldots N$-gram distributions $\mathcal{Y}_N(\mathcal{L}) : \Sigma^N \to \mathcal{R}$ is defined as:*

$$\mathcal{Y}_N(\mathcal{L}) = \alpha^{N-1} \cdot \mathcal{P}^c_1(\mathcal{F}_1[\mathcal{L}]) \cup \bigcup_{M=2}^{N} \left(\alpha^{N-M} \cdot \left(\mathcal{P}^{c,D}_M(\mathcal{F}_M[\mathcal{L}]) \cup \mathcal{E}^{1,\{\alpha\}}_M[\mathcal{P}^{c,R}_M(\mathcal{F}_M[\mathcal{L}])]\right) \right) \ .$$

The base part of $\mathcal{Y}_N(\mathcal{L})$ is defined by the unigram distribution $\mathcal{P}^c_1(\mathcal{F}_1[\mathcal{L}])$ which is prefixed with $N-1$ α-symbols. Note that in the case of unigrams, conditional and joint distributions are the same. The other part of the unified distribution contains for every M (with $1 < M \leq N$) a sublanguage which is the union of two weighted subsets: first the discounted M-gram probability distribution $\mathcal{P}^{c,D}_M(\mathcal{F}_M[\mathcal{L}])$ and second the residual probability mass $\mathcal{P}^{c,R}_M(\mathcal{F}_M[\mathcal{L}])$. For the latter, the suffix expansion WRT $\mathcal{E}^{1,\{\alpha\}}_M$ ensures that it consists of words $w_1 \ldots w_{M-1} \cdot \alpha$ whose associated weight corresponds to the $\alpha(w_1^{M-1})$-value in Equation (7) and which is computed

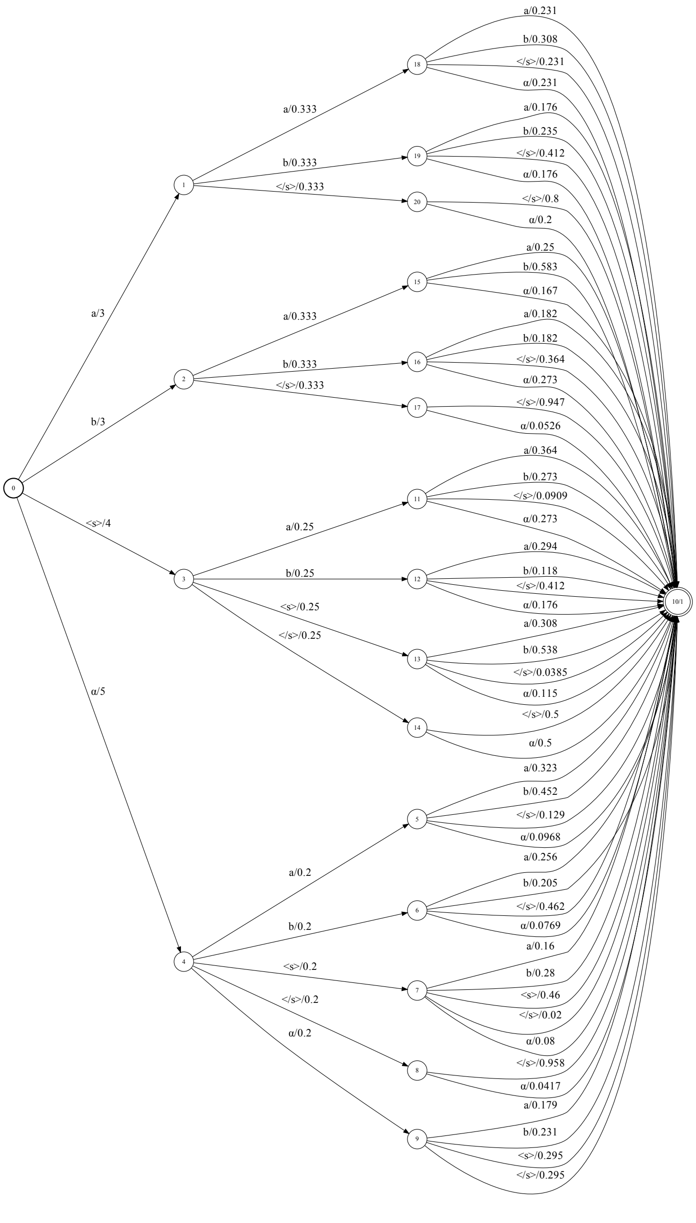

Figure 10: Unified distribution containing all $\{1, 2, 3\}$-gram subdistributions.

by the smoothing method. Note that the strings in $\mathcal{Y}_N(\mathcal{L})$ are by definition all of length N.

Fig. 10 shows the unified distribution for the trigrams of the example corpus.

Lemma 8 ($\mathcal{Y}_N$ defines a conditional probability distribution over $(\Sigma \cup \{\alpha\})^N$)**.**

Proof. All strings in $\mathcal{Y}_N$ are of length N and are either of the form $\alpha^{N-1}\Sigma$ (unigram case) or of the form $\alpha^{N-M}\Sigma^{M-1}(\alpha|\Sigma)$ (for $1 < M \leq N$) and originate from a single subset in Definition 30 since all those subsets are mutually disjoint. In the unigram case, for each symbol $a \in \mathcal{P}_1^c(\mathcal{F}_1[\mathcal{L}])$, $\alpha^{N-1}\mathcal{P}_1^c(\mathcal{F}_1[\mathcal{L}])$ is associated with the conditional probability $\Pr(a|\alpha^{N-1})$, since $\mathcal{P}_1^c(\mathcal{F}_1[\mathcal{L}])$ is a probability distribution by construction. By Lemma 7, the union of $\mathcal{P}_M^{c,D}$ and $\mathcal{P}_M^{c,R}$ gives a conditional probability distribution over $(\Sigma \cup \{\alpha\})^M$. Prefixing it with $N - M$ αs results in a conditional probability distribution over $(\Sigma \cup \{\alpha\})^N$. □

7.2.2 Back-off Navigation

Concerning the second point in the enumeration above, the possible sequences of M-grams according to Equation (7) have to be taken into account.

Example 3. Consider the trigram case and the input *abcde*, $c|ab$ has been processed, thus $d|bc$ is to be read next. If the trigram *bcd* and the bigram *cd* are not available we back-off successively to $d|c$ and to d. Now that d has been processed, e comes next. Since we already know that *cd* does not exist, concatenating $e|cd$ can not be correct. The correct continuation is $e|d$, the second case in Equation (9). This motivates why the w_i-transition from the ε-state in Figure 8 first traverses a bigram state before eventually going back to the trigram level.

Simply using the closure of $\mathcal{Y}_N$ as the input of the N-gram concatenator is thus not correct. Instead, we define a WRT called *back-off navigator* which ensures that incorrect sequences of M-grams are filtered from $(\mathcal{Y}_N)^*$.

Definition 31 (Back-off Navigator)**.** *A WRL* $\mathcal{B}_N : ((\Sigma \cup \{\alpha\})^N)^* \to \mathcal{R}$ *is defined for a finite alphabet* Σ *and the model parameter* N *as follows:*

$$\mathcal{B}_N = (\Sigma^N)^* \cup \mathcal{B}_{N-1,N} \ .$$

The back-off part $\mathcal{B}_{M,N}$ *(with* $0 \leq M < N$*) is recursively defined in the following way:*

$$\mathcal{B}_{M,N} = \begin{cases} \{\varepsilon\} & \text{if } M = 0 \\ \left(\Sigma^M \cdot \{\alpha \cdot \alpha^{N-M}\} \cdot \mathcal{B}_{M-1,N} \cdot \Sigma^M \cdot \{\alpha^{N-M-1}\}\right)^* & \text{if } M > 0 \ . \end{cases}$$

$\mathcal{B}_{M,N}$ accounts for the impossibility of recognizing a symbol in the $M+1$-subdistribution of an N-gram model ($0 < M < N$). This failure – indicated by α – may happen after having read M symbols. We then enter the nearest subdistribution which we find in $(\mathcal{Y}_N)^*$ after reading an α-prefix of length $N - M$.

Here, we may successfully process the M-gram or recursively back-off to the lower ordered distribution. In both cases – motivated by Example 3 and defined in Equation (9) – we continue processing in the nearest superdistribution which we find in $(\mathcal{Y}_N)^*$ after reading a prefix of $N - M - 1$ αs. Note that the length of all strings in $\mathcal{B}_N$ is a multiple of N.

Fig. 11 shows the trigram navigator implementing the back-off strategy for $N = 3$. Since we prefix the input to the model as well as the sentences of the training corpus with $N - 1$ delimiter symbols, failure can only occur after reading $N - 1$ symbols, because every suffix of an N-gram of length $N - 1$ also acts as a prefix of an N-gram (this can be easily shown by induction). This motivates why the back-off navigator in Figure 11 has α transitions only in state 2 (back-off from trigrams to bigrams) and state 5 (back-off to the unigrams). The remaining α-transitions serve to navigate to the nearest sub- (states 3, 6, 7) or superdistribution (state 9).

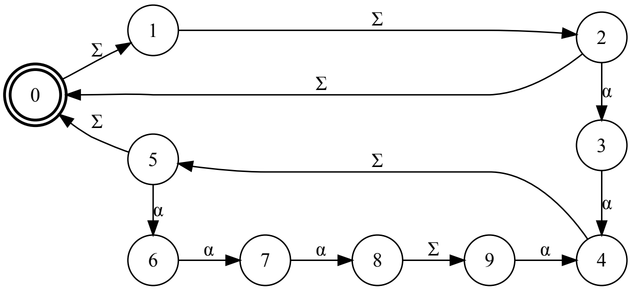

Figure 11: Back-off Navigator $\mathfrak{B}_3$.

Lemma 9 (Backoff-αs)**.** *Let $\mathcal{P}_M^{c,R}$ $(1 < M \leq N)$ be as defined in Definition 29. For each string w_1^M, $\big(\mathcal{E}_M^{1,\{\alpha\}}[\mathcal{P}_M^{c,R}]\big)(w_1^{M-1}\alpha)$ is equal to $\alpha(w_1^{M-1})$ in Equation (7).*

Proof. As defined in Equation (9), $\alpha(w_1^{M-1})$ is the residual probability mass computed by the discounting method for history w_1^{M-1}. By Lemma 6, $\mathcal{P}_M^{c,R}$ contains exactly that probability mass for all M-grams. By definition of application, $\big(\mathcal{E}_M^{1,\{\alpha\}}[\mathcal{P}_M^{c,R}]\big)(w_1^{M-1}\alpha)$ maps the sum of all conditional probabilities of all strings $w_1^{M-1}a$ for $a \in \Sigma$ to $w_1^{M-1}\alpha$. □

7.2.3 Robust Overlapping Concatenation

The overlapping concatenation $\cdot_N$ is the basis for the operator $\mathcal{D}_N$ which filters sequences of non-overlapping N-grams from the closure of all N-grams $(\Sigma^N)^*$. In parallel, a *robust* overlapping concatenation $\cdot_N^{\phi}$ is defined which allows the shortening and extension of histories during overlapping.

Definition 32 (Robust Overlapping Concatenation)**.** *The robust overlapping concatenation* $\mathcal{S} \cdot_N^{\alpha} \mathcal{Q}$ *of two weighted transductions* $\mathcal{S}$ *and* $\mathcal{Q}$ *is a mapping* $(\Sigma \cup \{\alpha\})^* \times (\Delta \cup \{\alpha\})^* \to \mathcal{R}$ *defined by*

$$\begin{aligned} \forall x \in \Sigma^*, \forall y \in \Delta^*,\ (\mathcal{S} \cdot_N^{\alpha} \mathcal{Q})(x,y) \ &= \ \mathcal{S}(x,y) \cdot_N \mathcal{Q}(x,y) \cup \\ &\bigoplus_{x=u\cdot v_1^{N-2}\cdot\alpha\cdot w, y=st} \ \bigcup_{i=1}^{N-1} \mathcal{S}\big(u \cdot v_1^{N-2} \cdot \alpha, s\big) \otimes \mathcal{Q}(\alpha^i \cdot v_i^{N-2} \cdot w, t)\ . \end{aligned}$$

$\cdot_N^{\alpha}$ successively increases the number of αs to be processed while shortening the N-gram history v_1^{N-2}.

Example 4. In the trigram case, Definition 32 boils down to the following cases for input $abc(d)$:[10]

$a \cdot bc$	$\cdot_N$	$bc \cdot d$	Normal, non-failure case
$a \cdot b\alpha$	$\cdot_N^{\alpha}$	$\alpha b \cdot c$	Processing in the 2-grams by shortening the history to b
$\alpha \cdot b\alpha$	$\cdot_N^{\alpha}$	$\alpha\alpha \cdot c$	Processing in the 1-grams by shortening the history to ε
$\alpha \cdot \alpha c$	$\cdot_N$	$\alpha c \cdot d$	1-grams → 2-grams

Cases 2 and 3 in Example 4 are distinguished from the others by the failure-indicating α at the last position of the first trigram. Note that the last case is handled by the standard overlapping mechanism if α is treated as a normal symbol in Σ.

Now, everything is prepared to define the WRT which repeatedly applies $\cdot_N^{\alpha}$ to an input string. The αs which trigger the shortening of the histories in Definition 32 are introduced by occurrences of failure symbols ϕ in the input string.

Definition 33 (Robust N-gram Concatenator $\mathcal{D}_N^{\phi}$)**.** *Let* $\mathcal{D}_N^{\alpha}$ *be as in Definition 22 with* $(\Sigma \cup \{\alpha\})$ *in place of* Σ *and* $\cdot_N^{\alpha}$ *instead of* $\cdot_N$*.* $\mathcal{D}_N^{\phi}$ *is a mapping* $(\Sigma \cup \{\alpha\})^* \times (\Sigma \cup \{\phi\})^* \to \mathcal{R}$ *defined by*

$$\mathcal{D}_N^{\phi} = \mathcal{D}_N^{\alpha} \circ (ID(\Sigma \setminus \{\alpha\}) \cup (\{\alpha\} \times \{\phi\}))^*\ .$$

Note that $\mathcal{D}_N^{\alpha}$ outputs – as before – only the last symbol of each N-gram, which may be α in the failure case (cf. Definition 22). $\mathcal{D}_N^{\phi}$ then simply replaces this occurrence of α by ϕ. Observe furthermore that Definition 33 is over-general, since it admits more αs than necessary. This over-generality is harmless since the sequences of αs and Σs are further constrained by the back-off navigator $\mathcal{B}_N$ (see Definition 34).

Fig. 12 shows the robust version of the trigram concatenator of Figure 6. Dashed transitions correspond to backing-off to the lower bigram and unigram distributions. Note that the actual implementation of $\mathcal{D}_N^{\phi}$ (see Figure 12) uses a weaker equiv-

[10] These cases are also the base of the proof of Theorem 2 (cf. Section 7.3).

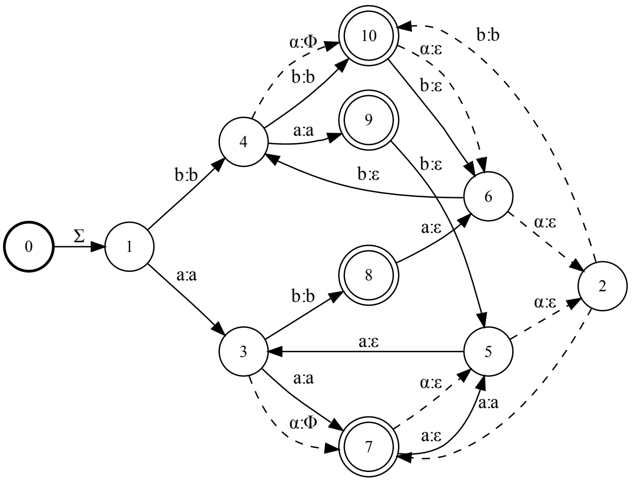

Figure 12: Robust trigram concatenator for $\Sigma = \{a, b\}$. The dashed transitions account for the back-off cases.

alence relation with respect to the states' *right relation.*[11] The implementation merges some non-equivalent states to allow for a compact representation of $\mathcal{D}_N^\phi$ which only differs minimally from the non-robust counterpart (following the back-off scheme, we would have to split for example state 10 in Figure 12 into two states to distinguish between the two possible continuations after having failed with the last input symbol b or successfully processed it. The concatenator in Figure 12 thus accepts for example the sequence $bbb\alpha\alpha b$ which is not admissible after the back-off scheme in Figure 11). Again, this coarsening is harmless because of the filter $\mathcal{B}_N$.

7.3 Putting It All Together

The back-off language model is obtained by applying $\mathcal{B}_N$, $\mathcal{U}_N^\alpha$ and $\mathcal{D}_N^\phi$ to the unified distribution.

Definition 34 (Robust language model)**.** *Let $\mathcal{L}$ be a weighted language over Σ^*. Let $\mathcal{U}_N^\alpha$ be the N-gram unfolder of Definition 23 where $(\Sigma \cup \{\alpha\})$ is used in place of Σ. The* robust language model *$\mathcal{M}_N^\phi(\mathcal{L})$ is a WRT $\Sigma^* \to \mathcal{P}, w \in \Sigma^* \mapsto \hat{\Pr}(w)$:*

$$\mathcal{M}_N^\phi(\mathcal{L}) = \mathcal{D}_N^\phi\left[\mathcal{Y}_N(\mathcal{L})^* \cap \mathcal{U}_N^\alpha \cap \mathcal{B}_N\right] .$$

[11] The *right relation* of a state q in a WFST $\mathcal{T}$ (*right language* in the case of WFSAs) is the WRT accepted by $\mathcal{T}$ when q is taken as the start state. Two states are equivalent (and can thus be merged during minimisation), if they have the same right relation.

We assume a mechanism which introduces the special failure symbol ϕ at the "right" places in w. Since the interpretation of ϕ is procedural in nature, we delegate this task to a special WFSA intersection algorithm. Note that the length of ϕ is 0.

From a procedural viewpoint, Definition 34 works in the following way:

- Each symbol in the input – a "normal" symbol $a \in \Sigma$ or ϕ – triggers a full cycle through $\mathcal{Y}_N$. Symbols $a \in \Sigma$ are mapped to themselves while ϕ is mapped to the appropriate $\alpha()$-value of the back-off Equation (7).
- An occurrence of ϕ is found in the input w (actually, the places where ϕ can occur are constrained by $\mathcal{Y}_N$).
- ϕ is mapped to α in the upper part of the relation by Definition 33.
- This α triggers other αs to be inserted into the relation's upper part by Definition 32.
- These additional αs determine the correct subdistribution in $\mathcal{Y}_N^*$ including the determination of the correct $\alpha()$-values of Equation (7).
- In addition, these αs are subject to the filtering of the back-off navigator $\mathcal{B}_N$ which also handles the navigation to the correct superdistribution after having read a number of ϕs.

Since complete trigram models tend to be large and our focus lies on the demonstration of the back-off mechanism, we depart from our previous example. Fig. 13 shows a back-off model following Definition 34 on the basis of the corpus `a|baaaa|baaaa`. For a better understanding of this example, the states are labeled with their histories. Also, the two states corresponding to the initial <s>-prefix have been deleted.

Theorem 2 (Robust language model $\mathcal{M}_N^{\phi}$)**.** *Given $\mathcal{M}_N^{\phi}(\mathcal{L})(w)$ as defined in Definition 34, $\mathcal{M}_N^{\phi}(\mathcal{L})(w)$ computes the correct probability for a delimited input string w after equations (20), (7) and (9).*

Proof. See Appendix A. □

7.4 Implementation and Complexity

The observations of Section 6.2 carry over to the back-off case. Of course, the intersection of $\mathcal{U}_N^{\alpha}$ and $\mathcal{B}_N$ in Definition 34 can be done by a virtual intersection algorithm. Due to their sizes, all three WRTs should be virtually constructed as well.

The application of the language model $\mathcal{M}_N^{\phi}$ to a (delimited) input string w is as usual the intersection of the trivially weighted WFSA for w with $\mathfrak{M}_N^{\phi}$, the WFSA corresponding to $\mathcal{M}_N^{\phi}$. Since $\mathfrak{M}_N^{\phi}$ contains transitions labeled with the special failure symbol ϕ, the normal intersection algorithm must be augmented with a mechanism which treats ϕ as a conditional ε-transition.

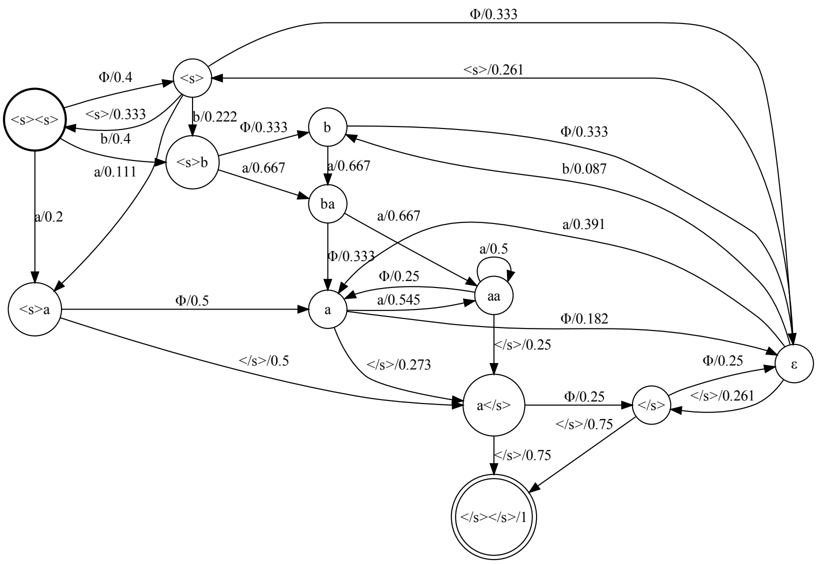

Figure 13: A back-off trigram model for the corpus $(a|baaaa|baaaa)$.

8 Conclusion and Further Work

In the previous sections, we have tried to show – to our knowledge for the first time – that the construction of language models can be defined on the formal language level alone without resorting to algorithms which manipulate the underlying WFSAs on a state level. We therefore make use of certain semiring properties, as for example divisibility, to model arithmetic statements like discounting procedures within the algebra of WRLs – an approach which may be applied to many other problems in the field of language processing.

Our formalisation is modular and can be seen as generator – $\mathcal{Y}_N^*$ – and a sequence of filters:

- The unified distribution $\mathcal{Y}_N$ accounts for the probabilities and combines discounted and residual probabilities for various values of M without caring about the back-off structure and specific N-gram histories.
- The N-gram unfolder ensures the macro structure of the model (cf. [14], p.83) with the delimiters at the beginning and end of each sentence.
- The back-off navigator reproduces the back-off strategy at a very general

level, again without distinguishing specific N-gram histories or calculating probabilities.

- Finally, the robust overlapping concatenation mechanism provides a correct handling of N-gram histories of various length by filtering out illegal ones.

We gave hints for efficient implementations of the five auxiliary WRTs which depend only on Σ and N. All steps are already implemented in the framework of [12].[12]

As previously pointed out, our work is primarily a theoretical one. The huge intermediate automata may prevent its practical application for corpora of the sizes currently used in NLP. Future work will therefore have to concentrate on mechanisms which allow the creation of individual language models for small parts of the underlying corpus and their subsequent combination. In addition, we currently investigate parallel versions of the automata algorithms which exploit multi-processor technology now available.

Another task is the reformulation of state-of-the-art discounting and smoothing methods and the clarification of the relationship between back-off and the other important strategy – interpolation – on a language-theoretic level.

Acknowledgments

We would like to thank Heiko Vogler and our two anonymous reviewers for their critical and very constructive remarks. We are also grateful to Sina Zarrieß, Sebastian Maar and Johannes Bubenzer for very valuable discussions.

References

[1] Aho, Alfred V. and Corasick, Margaret J. Efficient String Matching: An Aid to Bibiographic Search. *Communications of the Asscociation for Computing Machinery*, 18(6):333–340, 1975.

[2] Allauzen, Cyril, Mohri, Mehryar, and Roark, Brian. Generalized Algorithms for Constructing Statistical Language Models. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, volume 41, pages 40–47. The Association for Computational Linguistics, 2003.

[3] Bahl, Lalit R., Jelinek, Frederick, and Mercer, Robert L. A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Pami-5(2):179–190, 1983.

[4] Bakewell, Adam and Ghica, Dan R. On-the-Fly Techniques for Game-Based Software Model Checking. In Ramakrishnan, C.R. and Rehof, Jakob, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, volume

[12] All figures in the article were automatically generated by scripts which were processed by the framework's interpreter $fsm2$.

4963 of *Lecture Notes in Computer Science*, pages 78–92. Springer, Berlin, Heidelberg, 2008.

[5] Bullen, Richard H., Jr and Millen, Jonathan K. Microtext: The Design of a Microprogrammed Finite State Search Machine for Full-Text Retrieval. In *Proceedings of the Fall Joint Computer Conference*, AFIPS Joint Computer Conferences, pages 479–488, New York, NY, 1972. ACM.

[6] Chen, Stanley F. and Goodman, Joshua. An Empirical Study of Smoothing Techniques for Language Modeling. Technical Report TR-10-98, Center for Research in Computing Technology, Harvard University, Cambridge, MA, 1998.

[7] de Bruijn, Nicolaas Govert. A Combinatorical Problem. *Proceedings of the Section of Sciences of the Koninklijke Nederlandse Akademie van Wetenschappen*, 49:758–764, 1946.

[8] Droste, Manfred and Zhang, Guo-Qiang. On Transformations of Formal Power Series. *Information and Computation*, 184(2):369–383, 2003.

[9] Eisner, Jason. Simpler and More General Minimization for Weighted Finite-State Automata. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, volume 1, pages 64–71, Morristown, NJ, 2003. Association for Computational Linguistics.

[10] Ésik, Zoltán and Kuich, Werner. Equational Axioms for a Theory of Automata. In Vide, Carlos Martín, Mitrana, Victor, and Păun, Gheorghe, editors, *Formal Languages and Applications*, volume 148 of *Studies in Fuzziness and Soft Computing*, chapter 10, pages 183–196. Springer, Berlin, Heidelberg, 2004.

[11] Geyken, Alexander. The DWDS Corpus: A Reference Corpus for the German Language of the 20th Century. In Fellbaum, Christiane, editor, *Collocations and Idioms: Linguistic, Lexicographic, and Computational Aspects*. Continuum Press, London, 2006.

[12] Hanneforth, Thomas. FSM<2.0> – C++ Library for Manipulating (Weighted) Finite Automata. http://www.ling.uni-potsdam.de/˜tom/fsm/, 2004.

[13] Hanneforth, Thomas. A Memory Efficient Epsilon-Removal Algorithm for Weighted Acyclic Finite-State Automata. In Piskorski, Jakub, Watson, Bruce, and Yli-Jyrä, Anssi, editors, *Finite-State Methods and Natural Language Processing - Post-proceedings of the 7th International Workshop FSMNLP 2008*, Frontiers in Artificial Intelligence and Applications, 191, pages 72 – 81, Amsterdam, 2008. IOS Press.

[14] Jelinek, Frederick. *Statistical Methods for Speech Recognition.* Language, Speech and Communication. MIT Press, Cambridge, MA, 1997.

[15] Jelinek, Frederick and Mercer, Robert L. Interpolated Estimation of Markov Source Parameters from Sparse Data. In Gelsema, Edzard S. and Kanal, Laveen N., editors, *Pattern Recognition in Practice*, pages 381–397. North-Holland Publishing Company, Amsterdam, 1980.

[16] Jurafsky, Daniel and Martin, James H. *Speech and Language Processing.* Prentice Hall Series in Artificial Intelligence. Prentice Hall, Upper Saddle River, NJ, 2000.

[17] Katz, Slava M. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(3):400–401, 1987.

[18] Kucera, Henry and Francis, W. Nelson. *Computational Analysis of Present-day American English.* Brown University Press, Providence, RI, 1967.

[19] Kuich, Werner and Salomaa, Arto. *Semirings, Automata, Languages*, volume 5 of *EATCS Monographs on Theoretical Computer Science.* Springer, Berlin, Heidelberg, 1986.

[20] Markov, Andrey A. An Example of Statistical Investigation in the Text of 'Eugene Onyegin' Illustrating Coupling of Tests in Chains. *Proceedings of the Academy of Science St. Petersburg*, 7:153–162, 1913.

[21] Mohri, Mehryar. Finite-State Transducers in Language and Speech Processing. *Computational Linguistics*, 23(2):269–311, 1997.

[22] Mohri, Mehryar. Minimization Algorithms for Sequential Transducers. *Theoretical Computer Science*, 234:177–201, 2000.

[23] Mohri, Mehryar. Generic Epsilon-Removal and Input Epsilon-Normalization Algorithms for Weighted Transducers. *International Journal of Foundations of Computer Science*, 13(1):129–143, 2002.

[24] Mohri, Mehryar. Semiring Frameworks and Algorithms for Shortest-Distance Problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321–350, 2002.

[25] Mohri, Mehryar and Riley, Michael D. A Weight Pushing Algorithm for Large Vocabulary Speech Recognition. In *European Conf. on Speech Communication and Technology, Aalborg, Denmark, Sep. 2001*, pages 1603–1606, 2001.

[26] Pereira, Fernando C.N. and Riley, Michael D. Speech Recognition by Composition of Weighted Finite Automata. In Roche, Emmanuel and Schabes, Yves, editors, *Finite-State Language Processing*, volume 12 of *Language, Speech, and Communication*, chapter 15, pages 433–453. The MIT Press, Cambridge, MA, 1997.

[27] Revuz, Dominique. Minimisation of Acyclic Deterministic Automata in Linear Time. *Theoretical Computer Science*, 92(1):181 – 189, 1992.

[28] Riley, Michael D., Pereira, Fernando C., and Mohri, Mehryar. Transducer Composition for Context-Dependent Network Expansion. In Kokkinakis, George, Fakotakis, Nikos, and Dermatas, Evangelos, editors, *EUROSPEECH '97 - 5th European Conference on Speech Communication and Technology*, pages 1427–1430. ISCA, 1997.

[29] Shannon, Claude Elwood. Prediction and Entropy of Printed English. *Bell Labs Technical Journal*, 30:50–64, 1951.

[30] Witten, Ian H. and Bell, Timothy C. The Zero Frequency Problem: Estimating the Probability of Novel Events in Adaptive Text Compression. *IEEE Transactions on Information Theory*, 37(4):1085–1094, 1991.

A Proofs

Lemma 1 (Correctness of conditional N-gram probabilisation)**.** *Definition 19 computes the conditional probability of each N-gram as a special case of Equation (14) (with $i = N$):*

$$\Pr(w_N | w_1^{N-1}) = \frac{\mathrm{C}(w_1^N)}{\sum\limits_{a \in \Sigma} \mathrm{C}(w_1^{N-1} \cdot a)} \ . \tag{21}$$

Proof.

$$\begin{aligned}
&\mathcal{E}_N^1[\mathcal{C}_N](w_1^N) \\
&= \bigoplus_{x \in \Sigma^*} \big(\mathcal{C}_N(x) \otimes \mathcal{E}_N^1(x, w_1^N)\big) && \text{def. of application} \\
&= \bigoplus_{x \in \Sigma^*} \big(\mathcal{C}_N(x) \otimes (ID(\Sigma^{N-1}) \cdot (\Sigma \times \Sigma))(x, w_1^N)\big) && \text{def. of } \mathcal{E}_N^1 \\
&= \bigoplus_{a \in \Sigma} \big(\mathcal{C}_N(w_1^{N-1} \cdot a) \otimes \Sigma^{N-1}(w_1^{N-1}) \otimes (\Sigma \times \Sigma)(a, w_N)\big) && \text{def. ID, } \cdot \\
&= \bigoplus_{a \in \Sigma} \big(\mathcal{C}_N(w_1^{N-1} \cdot a) \\
&\qquad \otimes \{(w_1^{N-1})\}(w_1^{N-1}) \otimes \Sigma(a) \otimes \Sigma(w_N)\big) && \text{def. of } \cup \text{ and } \times \\
&= \bigoplus_{a \in \Sigma} \big(\mathcal{C}_N(w_1^{N-1} \cdot a) \\
&\qquad \otimes \{(w_1^{N-1})\}(w_1^{N-1}) \otimes \{a\}(a) \otimes \{w_N\}(w_N)\big) && \text{def. of } \cup \\
&= \bigoplus_{a \in \Sigma} \big(\mathcal{C}_N(w_1^{N-1} \cdot a) \otimes \overline{1}\big) && \text{def. of singleton} \\
&= \bigoplus_{a \in \Sigma} \mathcal{C}_N(w_1^{N-1} \cdot a) && \text{neutral element}
\end{aligned}$$

Since both operands of the intersection in Definition 19 have the same language projection, $\otimes$-negation replaces each weight of an N-gram by its multiplicative inverse, and intersection $\otimes$-multiplies weights, Definition 19 mimics Equation (15). □

Lemma 2 (Correspondence of T and $\mathcal{T}_N$)**.** *Given a WRL* $\mathcal{L} : \Sigma^N \to \mathcal{R}$, $\forall w_1^N \in \Sigma^N : \mathcal{T}_N(\mathcal{L})(w_1^N) = \mathrm{T}(w_1^N)$

Proof.

$$
\begin{aligned}
&\mathcal{T}_N(\mathcal{L})(w_1^N) && \\
=&\mathcal{E}_N^1[\pi^L(\mathcal{L})](w_1^N) && \text{def. 26} \\
=&\bigoplus_{x\in\Sigma^*} \big(\pi^L(\mathcal{L})(x) \otimes \mathcal{E}_N^1(x, w_1^N)\big) && \text{def. of application} \\
=&\bigoplus_{x\in\Sigma^*} \big(\pi^L(\mathcal{L})(x) \otimes (ID(\Sigma^{N-1}) \cdot (\Sigma \times \Sigma))(x, w_1^N)\big) && \text{def. of } \mathcal{E}_N^1 \\
=&\bigoplus_{a\in\Sigma} \big(\pi^L(\mathcal{L})(w_1^{N-1} \cdot a) \otimes \Sigma^{N-1}(w_1^{N-1}) \otimes (\Sigma \times \Sigma)(a, w_N)\big) && \text{def. ID, } \cdot \\
=&\bigoplus_{a\in\Sigma} \big(\pi^L(\mathcal{L})(w_1^{N-1} \cdot a) && \\
&\otimes \{(w_1^{N-1})\}(w_1^{N-1}) \otimes \Sigma(a) \otimes \Sigma(w_N)\big) && \text{def. of } \cup \text{ and } \times \\
=&\bigoplus_{a\in\Sigma} \big(\pi^L(\mathcal{L})(w_1^{N-1} \cdot a) && \\
&\otimes \{(w_1^{N-1})\}(w_1^{N-1}) \otimes \{a\}(a) \otimes \{w_N\}(w_N)\big) && \text{def. of } \cup \\
=&\bigoplus_{a\in\Sigma} \big(\pi^L(\mathcal{L})(w_1^{N-1} \cdot a)\big) && \text{def. singleton, } \overline{1} \\
=&\bigoplus_{a\in\Sigma} \begin{cases} \overline{1} & \text{if } \mathcal{L}(w_1^{N-1} \cdot a) \neq \overline{0} \\ \overline{0} & \text{otherwise} \ . \end{cases} && \text{def. } \ \pi^L
\end{aligned}
$$

□

Lemma 4 (Reconstruction of Witten-Bell Discounting)**.** *Given a WRL* $\mathcal{C}_N : \Sigma^N \to \mathcal{R}$, $w_1^N \mapsto \mathrm{C}(w_1^N)$, $\mathcal{W}_N^D(\mathcal{C}_N)(w_1^N)$ *maps an N-gram to its Witten-Bell discounted frequency* $\tilde{\mathrm{C}}(w_1^N)$*.*

Proof.

$$
\begin{aligned}
&\mathcal{W}_N^D(\mathcal{C}_N)(w_1^N) && \\
=&\Big(\mathcal{C}_N \cap \big(\mathcal{N}_N(\mathcal{C}_N) \cap \big(\mathcal{N}_N(\mathcal{C}_N) \cup \mathcal{T}(\mathcal{C}_N)\big)^{-\overline{1}}\big)\Big)(w_1^N) && \text{def. 28} \\
=&\mathcal{C}_N(w_1^N) \otimes \big(\mathcal{N}_N(\mathcal{C}_N)(w_1^N) \otimes \big(\mathcal{N}_N(\mathcal{C}_N)(w_1^N) \oplus \mathcal{T}(\mathcal{C}_N)(w_1^N)\big)^{-\overline{1}}\big) && \text{def. of } \cup \text{ and } \cap \\
=&\,\mathrm{C}(w_1^N) \otimes \big(\mathrm{N}(w_1^N) \otimes \big(\mathrm{N}(w_1^N) \oplus \mathrm{T}(w_1^N)\big)^{-\overline{1}}\big) && \text{def. of C, N, T}
\end{aligned}
$$

Since $a^{-\overline{1}}$ is $\frac{1}{a}$ in the probability semiring, the last line is equal to Equation (5). The proof for $\mathcal{W}_N^R$ is constructed in the same manner. □

Lemma 5 (Witten-Bell Decomposition)**.** *Given a WRL $\mathcal{L} : \Sigma^N \to \mathcal{R}$, $\mathcal{W}_N^D(\mathcal{L}) \cup \mathcal{W}_N^R(\mathcal{L}) = \mathcal{L}$.*

Proof.

$$
\begin{aligned}
&\mathcal{W}_N^D(\mathcal{L}) \cup \mathcal{W}_N^R(\mathcal{L}) && \\
=&\big(\mathcal{L} \cap (\mathcal{N}_N(\mathcal{L}) \cap (\mathcal{N}_N(\mathcal{L}) \cup \mathcal{T}_N(\mathcal{L}))^{-\bar{1}})\big) \cup && \\
&\big(\mathcal{L} \cap (\mathcal{T}_N(\mathcal{L}) \cap (\mathcal{N}_N(\mathcal{L}) \cup \mathcal{T}_N(\mathcal{L}))^{-\bar{1}})\big) && \text{def. of } \mathcal{W}_N^D,\, \mathcal{W}_N^R \\
=&\big((\mathcal{L} \cap \mathcal{N}_N(\mathcal{L})) \cap (\mathcal{N}_N(\mathcal{L}) \cup \mathcal{T}_N(\mathcal{L}))^{-\bar{1}}\big) \cup && \\
&\big((\mathcal{L} \cap \mathcal{T}_N(\mathcal{L})) \cap (\mathcal{N}_N(\mathcal{L}) \cup \mathcal{T}_N(\mathcal{L}))^{-\bar{1}}\big) && \text{assoc. of } \cap \\
=&((\mathcal{L} \cap \mathcal{N}_N(\mathcal{L})) \cup (\mathcal{L} \cap \mathcal{T}_N(\mathcal{L}))) \cap (\mathcal{N}_N(\mathcal{L}) \cup \mathcal{T}_N(\mathcal{L}))^{-\bar{1}} && \cap \succ \cup \\
=&\mathcal{L} \cap \big((\mathcal{N}_N(\mathcal{L}) \cup \mathcal{T}_N(\mathcal{L})) \cap (\mathcal{N}_N(\mathcal{L}) \cup \mathcal{T}_N(\mathcal{L}))^{-\bar{1}}\big) && \cap \succ \cup \\
=&\bigoplus_{x \in \mathcal{L}} \bigg(\mathcal{L}(x) \otimes && \\
&\big((\mathcal{N}_N(\mathcal{L})(x) \oplus \mathcal{T}_N(\mathcal{L})(x)) \otimes (\mathcal{N}_N(\mathcal{L})(x) \oplus \mathcal{T}_N(\mathcal{L})(x))^{-\bar{1}}\big)\bigg) && \text{def. of } \mathcal{L},\, \cup,\, \cap \\
=&\bigoplus_{x \in \mathcal{L}} \big(\mathcal{L}(x) \otimes \bar{1}\big) && \text{def. of } {}^{-\bar{1}} \\
=&\mathcal{L} && \text{def. of } \mathcal{L},\, \bar{1}
\end{aligned}
$$

□

Lemma 7 (Union of $\mathcal{P}_N^{c,D}$ and $\mathcal{P}_N^{c,R}$)**.** *Let $\mathcal{L}$ denote a WRL $\Sigma^N \to \mathcal{R}$:*

$$\mathcal{P}_N^{c,D}(\mathcal{L}) \cup \mathcal{P}_N^{c,R}(\mathcal{L}) = \mathcal{P}_N^c(\mathcal{L}) \ .$$

Proof.

$$
\begin{aligned}
&\mathcal{P}_N^{c,D}(\mathcal{L}) \cup \mathcal{P}_N^{c,R}(\mathcal{L}) && \\
&\qquad = (\mathcal{W}_N^D(\mathcal{L}) \cap (\mathcal{N}_N(\mathcal{L}))^{-\bar{1}}) \cup (\mathcal{W}_N^R(\mathcal{L}) \cap (\mathcal{N}_N(\mathcal{L}))^{-\bar{1}}) && \text{by def. 29} \\
&\qquad = (\mathcal{W}_N^D(\mathcal{L}) \cup \mathcal{W}_N^R(\mathcal{L})) \cap (\mathcal{N}_N(\mathcal{L}))^{-\bar{1}} && \text{by } \cap \succ \cup \\
&\qquad = \mathcal{L} \cap (\mathcal{N}_N(\mathcal{L}))^{-\bar{1}} && \text{by lem. 5} \\
&\qquad = \mathcal{L} \cap (\mathcal{E}_N^1[\mathcal{L}])^{-\bar{1}} && \text{by def. 27} \\
&\qquad = \mathcal{P}_N^c(\mathcal{L}) && \text{by def. 19}
\end{aligned}
$$

□

Theorem 2 (Robust language model $\mathcal{M}_N^\phi$)**.** *Given $\mathcal{M}_N^\phi(\mathcal{L})(w)$ as defined in Definition 34, $\mathcal{M}_N^\phi(\mathcal{L})(w)$ computes the correct probability for a delimited input string w after equations (20), (7) and (9).*

Proof. Introductory remarks:

- For reasons of simplicity, we restrict the proof to the case of $N = 3$. Proofs for other values for N are analogous.
- Observe that the length of all strings in $\mathcal{Y}_N(\mathcal{L})^*$, $\mathcal{U}_N^\alpha$, $\mathcal{B}_N$ and $\pi^1(\mathcal{D}_N^\phi)$ is a multiple of N. The back-off navigator $\mathcal{B}_N$ places the most specific constraints on the form of the strings to which $\mathcal{D}_N^\phi$ is applied. Thus, in the trigram case, the shortest strings in $\mathcal{Y}_3(\mathcal{L})^* \cap \mathcal{B}_3$ (besides ε, which is ruled out by $\mathcal{U}_N^\alpha$) are of one of the following forms: Σ^3 or $\Sigma^2\alpha \cdot \alpha\Sigma^2$ or $\Sigma^2\alpha \cdot \alpha\Sigma\alpha \cdot \alpha^2\Sigma \cdot \alpha\Sigma^2$.
- Note that the length of ϕ is 0.
- For the reader's better understanding, we spell out the three different cases of Equation (7) for $N = 3$:

$$\hat{\Pr}(w_i|w_{i-2}^{i-1}) = \begin{cases} \tilde{\Pr}(w_i|w_{i-2}^{i-1}) & \text{if } \mathrm{C}(w_{i-2}^{i}) > 0 \\ \alpha(w_{i-2}^{i-1}) \cdot \tilde{\Pr}(w_i|w_{i-1}) & \text{if } \mathrm{C}(w_{i-2}^{i}) = 0 \;\&\; \mathrm{C}(w_{i-1}^{i}) > 0 \\ \alpha(w_{i-2}^{i-1}) \cdot \alpha(w_{i-1}) \cdot \Pr(w_i) & \text{otherwise .} \end{cases} \tag{22}$$

  Remember that the values of the αs in Equation (22) may be 1 in case the history is not present (cf. Equation (9)).

Since $\mathcal{U}_3^\alpha$ introduces the sentence delimiters, the proof is by induction on the length of the string $w = \text{<s>}^2\, w'\, \text{</s>}^2$.

Induction hypothesis:
Let $w_1^k = \text{<s>}^2\ w'\ \text{</s>}^2$ an input string of length $k \geq 4\ \ (= 2(N-1))$:

$$\mathcal{D}_3^\phi\big[\mathcal{Y}_3(\mathcal{L})^* \cap \mathcal{U}_3^\alpha \cap \mathcal{B}_3\big](w_1^k) = \prod_{i=3=N}^{k} \hat{\Pr}(w_i|w_{i-2}^{i-1}) \ . \tag{23}$$

Induction base: $|w| = 0$.
Case 1a: $w = \varepsilon$ (this means that the trigram $\text{<s>}^2\text{</s>}$ is in $\mathcal{Y}_3(\mathcal{L})$)

$$\begin{aligned}
&\mathcal{D}_3^\phi\big[(\mathcal{Y}_3)^*\big](\text{<s>}^2\text{</s>}^2) && \\
=&\bigoplus_{x\in\Sigma^*}\big((\mathcal{Y}_3^*)(x)\ \otimes\ \mathcal{D}_3^\phi(x, \text{<s>}^2\text{</s>}^2)\big) && \text{def. of appl.} \\
=&(\mathcal{Y}_3^*)(\text{<s>}^2\text{</s>} \cdot \text{<s>}\text{</s>}^2) && \\
&\otimes \big(\Sigma^3(\text{<s>}^2\text{</s>}) && \\
&\otimes \{(\text{<s>}\text{</s>}^2, \text{</s>})\}(\text{<s>}\text{</s>}^2, \text{</s>})\big) && \text{def. of } \mathcal{D}_3^\phi \\
=&(\mathcal{Y}_3^*)(\text{<s>}^2\text{</s>} \cdot \text{<s>}\text{</s>}^2) && \text{ID, singleton, } \overline{1} \\
=&(\mathcal{Y}_3)(\text{<s>}^2\text{</s>}) \otimes (\mathcal{Y}_3)(\text{<s>}\text{</s>}^2) && \text{closure} \\
=&(\mathcal{P}_3^{c,D})(\text{<s>}^2\text{</s>}) \otimes (\mathcal{P}_3^{c,D})(\text{<s>}\text{</s>}^2) && \text{def. of } \mathcal{Y}_3
\end{aligned}$$

$$\begin{array}{ll}
=\tilde{\Pr}(\text{</s>}\,|\,\text{<s>}^2) \otimes \tilde{\Pr}(\text{</s>}\,|\,\text{<s></s>}) & \text{lem. 6} \\
=\hat{\Pr}(\text{</s>}\,|\,\text{<s>}^2) \otimes \hat{\Pr}(\text{</s>}\,|\,\text{<s></s>}) & \text{eqs. (20), (22, case 1)}
\end{array}$$

Case 1b: $w = \phi\phi$ (this means that the trigram $\text{<s>}^2\text{</s>}$ is not in $\mathcal{Y}_3(\mathcal{L})$, which in turn entails that the trigram α <s></s> will also not be present in $\mathcal{Y}_3(\mathcal{L})$). $\mathcal{D}_3^{\phi}$ will decompose $\text{<s>}^2\,\phi\phi\,\text{</s>}^2$ into

$$(\text{<s>}^2\,\alpha, \text{<s>}^2\,\phi) \cdot (\alpha\,\text{<s>}\,\alpha, \phi) \cdot (\alpha^2\,\text{</s>}, \text{</s>}) \cdot (\alpha\,\text{</s>}^2, \text{</s>})$$

whose first projection is also in $\mathcal{U}_3^{\alpha} \cap \mathcal{B}_3$.

$$\begin{array}{ll}
\mathcal{Y}_3^*(\text{<s>}^2\,\alpha \cdot \alpha\,\text{<s>}\,\alpha \cdot \alpha^2\,\text{</s>} \cdot \alpha\,\text{</s>}^2) & \\
=\mathcal{Y}_3(\text{<s>}^2\,\alpha) \otimes \mathcal{Y}_3(\alpha\,\text{<s>}\,\alpha) \otimes \mathcal{Y}_3(\alpha^2\,\text{</s>}) \otimes \mathcal{Y}_3(\alpha\,\text{</s>}^2) & \text{closure} \\
=\mathcal{P}_3^{c,R}(\text{<s>}^2\,\alpha) \otimes \mathcal{P}_2^{c,R}(\text{<s>}\,\alpha) \otimes \mathcal{P}_1^{c}(\text{</s>}) \otimes \mathcal{P}_2^{c,D}(\text{</s>}^2) & \text{def. 30} \\
=\big(\alpha(\text{<s>}^2) \otimes \alpha(\text{<s>}) \otimes \Pr(\text{</s>})\big) \otimes \tilde{\Pr}(\text{</s>}|\text{</s>}) & \text{lem. 9, lem. 6} \\
=\hat{\Pr}(\text{</s>}|\text{<s>}^2)\ \otimes \tilde{\Pr}(\text{</s>}|\text{</s>}) & \text{eq. (22, case 3)} \\
=\hat{\Pr}(\text{</s>}|\text{<s>}^2)\ \otimes \big(\bar{1} \otimes \tilde{\Pr}(\text{</s>}|\text{</s>})\big) & \text{neutr. element of } \otimes \\
=\hat{\Pr}(\text{</s>}|\text{<s>}^2) \otimes\ \hat{\Pr}(\text{</s>}|\text{<s></s>}) & \text{eqs. (9), (22, case 2)}
\end{array}$$

Induction step: Assume, the induction hypothesis holds for strings $w_1^i = \text{<s>}^2$ w' </s>^2 with $4\ (=2(N-1)) \leq i \leq k$. We show that it also holds for $k+1$.
For the proof, there are two possible cases concerning the history w_{k-1}^k of w_{k+1}:[13]

1. $w_1^k = w'' w_{k-1}^k$ or $w_1^k = w'' w_{k-1}\ \phi\ w_k$: the history w_{k-1}^k is present. Here we have three subcases, depending on in which distribution we successfully process w_{k+1}:

   a) $w_1^{k+1} = w_1^k w_{k+1}$: trigrams

   b) $w_1^{k+1} = w_1^k \phi w_{k+1}$: bigrams

   c) $w_1^{k+1} = w_1^k \phi\phi w_{k+1}$: unigrams

2. $w_1^k = w'' w_{k-1}\ \phi\phi\ w_k$: the history w_{k-1}^k is not present, since w_k was processed (after reading two occurrences of ϕ) in the unigram distribution. Here we have two subcases to consider, which can only occur after case 1c) above or 2b) below:

   a) $w_1^{k+1} = w_1^{k-1}\ \phi\phi\ w_k w_{k+1}$: bigrams (superdistribution)

   b) $w_1^{k+1} = w_1^{k-1}\ \phi\phi\ w_k\ \phi\ w_{k+1}$: unigrams

In the following, we give proofs for the 5 subcases mentioned above.

[13] *Remark*: With respect to the back-off navigator in Figure 11, this distinction is reflected in the particular state the navigator is after having processed w_k: In the first case, this state is 0, while it is 9 in the second. In the general case of an N-gram navigator, there will be $N-1$ such states, and in turn $N-1$ main cases to consider.

1. a) $\mathcal{D}_3^\phi$ maps w_{k+1} to w_{k-1}^{k+1}, which is also in $\mathcal{B}_3$.

$$\begin{aligned}
&\mathcal{Y}_3(w_{k-1}^{k+1}) & \\
=&\mathcal{P}_3^{c,D}(w_{k-1}^{k+1}) & \text{def. 30}\\
=&\tilde{\Pr}(w_{k+1}|w_{k-1}^{k}) & \text{lem. 6}\\
=&\hat{\Pr}(w_{k+1}|w_{k-1}^{k}) & \text{eq. (22, case 1)}
\end{aligned}$$

b) $\mathcal{D}_3^\phi$ maps ϕw_{k+1} to $w_{k-1}^{k}\alpha \cdot \alpha w_k^{k+1}$, which is also in $\mathcal{B}_3$.

$$\begin{aligned}
&\mathcal{Y}_3^*(w_{k-1}^{k}\alpha \cdot \alpha w_k^{k+1}) & \\
=&\mathcal{Y}_3(w_{k-1}^{k}\alpha) \otimes \mathcal{Y}_3(\alpha w_k^{k+1}) & \text{def. of closure}\\
=&\mathcal{P}_3^{c,R}(w_{k-1}^{k}\alpha) \otimes \mathcal{P}_2^{c,D}(w_k^{k+1}) & \text{def. 30}\\
=&\alpha(w_{k-1}^{k}) \otimes \tilde{\Pr}(w_{k+1}|w_k) & \text{lem. 9, lem. 6}\\
=&\hat{\Pr}(w_{k+1}|w_{k-1}^{k}) & \text{eq. (22, case 2)}
\end{aligned}$$

c) $\mathcal{D}_3^\phi$ maps $\phi\phi w_{k+1}$ to $w_{k-1}^{k}\alpha \cdot \alpha w_k \alpha \cdot \alpha^2 w_{k+1}$. This string is not in $\mathcal{B}_3$, but is a prefix of one of its strings, namely $w_{k-1}^{k}\alpha \cdot \alpha w_k\alpha \cdot \alpha^2 w_{k+1} \cdot \alpha w_k^{k+1}$. The "missing" suffix αw_k^{k+1} will be covered in case 2a) or 2b), which are the only possible cases following 1c).

$$\begin{aligned}
&\mathcal{Y}_3^*(w_{k-1}^{k}\alpha \cdot \alpha w_k\alpha \cdot \alpha^2 w_{k+1}) & \\
=&\mathcal{Y}_3(w_{k-1}^{k}\alpha) \otimes \mathcal{Y}_3(\alpha w_k\alpha) \otimes \mathcal{Y}_3(\alpha^2 w_{k+1}) & \text{def. of closure}\\
=&\mathcal{P}_3^{c,R}(w_{k-1}^{k}\alpha) \otimes \mathcal{P}_2^{c,R}(w_k\alpha) \otimes \mathcal{P}_1^{c}(w_{k+1}) & \text{def. 30}\\
=&\alpha(w_{k-1}^{k}) \otimes \alpha(w_k) \otimes \Pr(w_{k+1}) & \text{lem. 9, lem. 6}\\
=&\hat{\Pr}(w_{k+1}|w_{k-1}^{k}) & \text{eq. (22, case 3)}
\end{aligned}$$

2. The following subcases cover the "missing" suffix of case 1c) above.

a) $\mathcal{D}_3^\phi$ maps w_{k+1} to αw_k^{k+1}.

$$\begin{aligned}
&\mathcal{Y}_3^*(\alpha w_k^{k+1}) & \\
=&\mathcal{Y}_3(\alpha w_k^{k+1}) & \text{def. of closure}\\
=&\mathcal{P}_2^{c,D}(w_k^{k+1}) & \text{def. 30}\\
=&\tilde{\Pr}(w_{k+1}|w_k) & \text{lem. 6}\\
=&\overline{1} \otimes \tilde{\Pr}(w_{k+1}|w_k) & \text{eq. (9, case 2)}\\
=&\hat{\Pr}(w_{k+1}|w_{k-1}^{k}) & \text{eq. (22, case 2)}
\end{aligned}$$

b) $\mathcal{D}_3^\phi$ maps ϕw_{k+1} to $\alpha w_k\alpha \cdot \alpha^2 w_{k+1}$.

$$\mathcal{Y}_3^*(\alpha w_k\alpha \cdot \alpha^2 w_{k+1})$$

$$\begin{aligned}
&= \mathcal{Y}_3(\alpha w_k \alpha) \otimes \mathcal{Y}_3(\alpha^2 w_{k+1}) && \text{def. of closure}\\
&= \mathcal{P}_2^{c,R}(w_k \alpha) \otimes \mathcal{P}_1^c(w_{k+1}) && \text{def. 30}\\
&= \alpha(w_k) \otimes \Pr(w_{k+1}) && \text{lem. 9, lem. 6}\\
&= \overline{1} \otimes \alpha(w_k) \otimes \Pr(w_{k+1}) && \text{eq. (9, case 2)}\\
&= \hat{\Pr}(w_{k+1}|w_{k-1}^k) && \text{eq. (22, case 3)}
\end{aligned}$$

Combining this with the induction hypothesis, we get

$$\mathcal{D}_3^\phi\big[\mathcal{Y}_3(\mathcal{L})^* \cap \mathcal{U}_3^\alpha \cap \mathcal{B}_3\big](w_1^{k+1}) = \prod_{i=N}^{k+1} \hat{\Pr}(w_i|w_{i-2}^{i-1}) \; . \tag{24}$$

Note that the $N-1$ sentence delimiters </s> ensure that the N-grams α </s>$^{N-1}$ or a </s>$^{N-1}$ for some $a \in \Sigma$ are always present in $\mathcal{Y}_N$ such that the last step of the computation of the decomposed back-off probability of an delimited input sentence will always be case 1a) or case 2a). □

B Constructions

Definition 35 (Construction of $\mathfrak{F}_N$)**.** *The weighted finite state transducer $\mathfrak{F}_N$ wrt a semiring $\mathcal{R}$ is an 8-tuple $\langle Q, \Sigma, \Sigma \cup \{\varepsilon\}, 0, F, E_i \cup E_m \cup E_f, \overline{1}, \rho\rangle$ where*

$$\begin{aligned}
Q &= \bigcup_{i=0}^{N} \{i\}\\
F &= \{N\}\\
E_i &= \bigcup_{a\in\Sigma} \{(0,0,a,\varepsilon,\overline{1})\} \cup \bigcup_{a\in\Sigma} \{(0,1,a,a,\overline{1})\}\\
E_m &= \bigcup_{i=1}^{N-1} \bigcup_{a\in\Sigma} \{(i,i+1,a,a,\overline{1})\}\\
E_f &= \bigcup_{a\in\Sigma} \{(N,N,a,\varepsilon,\overline{1})\}\\
\forall q \in F, \rho(q) &= \overline{1}
\end{aligned}$$

Definition 36 (Construction of $\mathfrak{E}_N^k$)**.** *The weighted finite state transducer $\mathfrak{E}_N^k$ wrt* *a semiring $\mathcal{R}$ is an 8-tuple $\langle Q, \Sigma, \Sigma, 0, F, E_m \cup E_k, \overline{1}, \rho \rangle$ where*

$$\begin{aligned} Q &= \bigcup_{i=0}^{N} \{i\} \\ F &= \{N\} \\ E_m &= \bigcup_{i=0}^{N-k-1} \bigcup_{a \in \Sigma} \{(i, i+1, a, a, \overline{1})\} \\ E_k &= \bigcup_{i=N-k}^{N-1} \bigcup_{a \in \Sigma} \bigcup_{b \in \Sigma} \{(i, i+1, a, b, \overline{1})\} \\ \forall q \in F, \rho(q) &= \overline{1} \end{aligned}$$

Definition 37 (Construction of $\mathfrak{D}_N$)**.** *The weighted finite state transducer $\mathfrak{D}_N$ wrt a semiring $\mathcal{R}$ is an 8-tuple $\langle Q, \Sigma, \Sigma, 0, F, E_0 \cup E_t \cup E_o \cup E_l, \overline{1}, \rho \rangle$ where (using q_y from Definition 25):*[14]

$$Q = \bigcup_{i=0}^{q_y+|\Sigma|^{N-1}*(N-1)-1} \{i\}$$

$$F = \bigcup_{i=q_y}^{q_y+|\Sigma|^{N-1}-1} \{i\}$$

$$E_0 = \bigcup_{a \in \Sigma} \{(0, 1, a)\}$$

$$E_t = \bigcup_{i=1}^{q_y-1} \bigcup_{a \in \Sigma} \Big\{ \Big(i, \big(i * |\Sigma| + \mathtt{idx}(a) \big) - \big(|\Sigma| - 2 \big), a, a, \overline{1} \Big) \Big\}$$

$$E_o = \bigcup_{i=q_y}^{q_y+|\Sigma|^{N-1}*(N-2)-1} \Big\{ (i, i + |\Sigma|^{N-1}, a, \varepsilon, \overline{1}) \Big| \mathtt{idx}(a) = \left\lfloor \frac{i - q_y}{|\Sigma|^{N-2-\left\lfloor \frac{i-q_y}{|\Sigma|^{N-1}} \right\rfloor}} \right\rfloor \bmod |\Sigma| \Big\}$$

$$E_l = \bigcup_{i=q_y+|\Sigma|^{N-1}*(N-2)}^{q_y+|\Sigma|^{N-1}*(N-1)-1} \{(i, \mathtt{idx}(a) + 2, a, \varepsilon, \overline{1}) \big| \mathtt{idx}(a) = (i - q_y) \bmod |\Sigma|\}$$

$\rho(q) = \overline{1}, \forall q \in F.$

Received 15th August 2008

[14] $\lfloor x \rfloor$ denotes the *floor* value of a number. E.g. $\lfloor 2.34 \rfloor = 2$.