A Layered, Transducer-Based Model for Speech and Language Processing

Kallirroi Georgila, Kyriakos Sgarbas, Nikos Fakotakis, George Kokkinakis

Wire Communications Laboratory
Electrical and Computer Engineering Dept.
University Of Patras, 265 00 Patras, Greece
Tel. +30 2610 991722 Fax. +30 2610 991855

{rgeorgil, sgarbas ,fakotaki, gkokkin}@wcl.ee.upatras.gr

Abstract: This paper presents a transducer-based model for speech and language processing. The proposed model consists of a series of layers of interconnected transducers. At the lower layer there is a Finite-State Transducer (FST) containing the lexicon of the system. At the next layer another FST represents the language model. Then a word-to-POS transducer is used to provide a link between the graphemic form of the word and the part-of-speech tag associated with it. The upper layer is composed of a transducer, which utilises the POS information of the previous layer to form syntactic structures based on context-free grammatical rules. Transition probabilities are also considered thus forming Weighted Finite-State Transducers (WFSTs). Keeping these probabilities outside the grammatical information allows their independent composition. The grammatical component can be composed from existing carefully built lexicons, language models and syntactic rules, while the probabilities can be derived automatically from corpora afterwards and be applied as additional information to the existing structure. A set of on-line algorithms for rapid update of automata and transducers has been developed to support this approach.

INTRODUCTION

A Finite-State Transducer (FST) is a generalisation of a Finite-State Automaton (FSA). That is, a transducer is a finite-state device that encodes a mapping between input and output symbol sequences. Whereas a FSA has arcs bearing a single symbol and represents a (possibly infinite) set of symbol sequences, a FST has its arcs labelled with input and output symbols, and represents a set of pairs of {input symbol sequence, output symbol sequence}. In addition, a Weighted Finite-State Transducer (WFST) also has a cost associated with each arc or accepting state (1).

Since many information sources in speech and language processing involve stochastic finite-state mappings between symbol sequences, weighted transducers are a natural choice to represent them (2). Therefore, a WFST is a suitable representation for HMMs, context-dependency, pronunciation dictionaries, grammars, *n*-gram language models, and alternative recognition outputs (1)(3)(4)(5). Furthermore, many of the methods used to combine and optimise these information sources in speech and language processing can be efficiently implemented in terms of mathematically well-defined primitive operations on weighted transducers (5).

This paper presents a transducer-based model for speech recognition and syntactic parsing, and in the opposite direction for natural language generation and speech synthesis. The proposed model consists of a series of layers of interconnected transducers. At the lower layer there is a FST containing the lexicon of the system. At the next layer another FST represents the language model. Then a word-to-POS transducer is used to provide a link between the graphemic form of the word and the part-of-speech tag associated with it. The upper layer is composed of a transducer, which utilises the POS information of the previous layer to form syntactic structures based on context-free grammatical rules. Transition probabilities are also considered thus forming WFSTs. That is, a WFST is a generalisation of a Weighted Finite-State Automaton (WFSA). Keeping these probabilities outside the grammatical information allows their independent composition. The grammatical component can be composed from existing carefully built lexicons, language models and syntactic rules, while the probabilities can be derived automatically from corpora afterwards and be applied as additional information to the existing structure. A set of on-line algorithms for rapid update of automata and transducers has been developed to support this approach.

The paper is organized as follows: At first the definition of WFSAs and WFSTs is given together with the algorithm that is used for the construction of the WFSTs. Then it is described how WFSTs can formulate lexicons, language models and rules of context-free grammars. In the following, the combination of different types of WFSTs is discussed and ideas for future work are presented. Moreover, it is explained why the lexicon transducers should change form in order to be incorporated into the speech recognition process and in the opposite direction in speech synthesis. Finally, some conclusions are given in the final section.

WEIGHTED FINITE-STATE TRANSDUCERS

Network models such as HMMs, language models, etc., used in speech and language processing are special cases of Weighted Finite-State Automata (WFSAs). A WFSA is a 6-tuple (Q, i, F, A, P, δ) where

- Q is the set of states,
- $i \in Q$ is the initial state,
- $F \subseteq Q$, the set of final states,
- A is the finite set of input symbols,
- P is the set of probabilities corresponding to the state transitions,
- δ is the state transition function which maps $Q \times A \times P$ to Q.

A transition $t = (t_s, l(t), w(t), t_d)$ can be represented by an arc from the source state t_s to the destination state t_d, with the label $l(t)$ and weight $w(t)$. In speech and language processing, the transition weight $w(t)$ often represents a probability or a log probability (2)(5).

Weighted Finite-State Transducers (WFSTs) generalise WFSAs by replacing the single transition label by a pair (i, o) of an input label i and an output label o. While a weighted automaton associates symbol sequences and weights, a WFST associates pairs of symbol sequences and weights, that is, it represents a weighted binary relation between symbol sequences (6). Thus a WFST is an 8-tuple $(Q, i, F, A, B, P, \delta, \sigma)$ where Q, i, F, A, P, δ are the same as for WFSA, B is the finite set of output symbols and σ is the state transition function which maps $Q \times A \times P$ to B^*. A transition $t = (t_s, l_i(t), l_o(t), w(t), t_d)$ can be represented by an arc from the source state t_s to the destination state t_d, with the input label $l_i(t)$, the output label $l_o(t)$ and the weight $w(t)$.

In order to create the FSTs, we use an algorithm that was first developed for constructing Directed Acyclic Word Graphs (DAWGs) and then it was updated and improved to deal with WFSTs. A DAWG is a special case of a FSA where no loops (cycles) are allowed. We use incremental construction of WFSTs in order to be able to update them without having to build them from scratch in every change. We have applied the incremental construction algorithms described in (7)(8) which run in real-time.

LEXICON

At the lower layer of the proposed model there is a FST containing the lexicon of the system. Phone-based speech recognisers usually employ a dictionary, i.e. a list of word spellings together with their pronunciations, as a means of grapheme-to-phoneme conversion. In our approach, the FST simply encodes a dictionary lookup. Thus each word is stored as a directed path in the graph of the transducer. Both graphemic and phonetic transcriptions are stored on the transducer transitions at character/phoneme detail.

The procedure works as follows. Suppose the lexicon contains the Greek names:

γιάννης Y i a n i s

γιάννης Y a n i s

διονύσης D i o n i s i s

διονύσης D Y o n i s i s

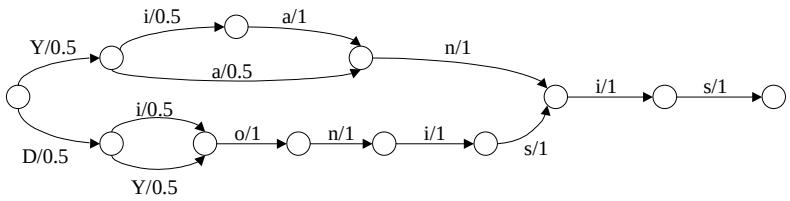
The first column gives the graphemic form of the word and the second its phonetic transcription. Note that there are words that have multiple phonetic transcriptions. By considering only the phonetic form of the words, the input to our DAWG construction algorithm are the strings *Y i a n i s*, *Y a n i s*, *D i o n i s i s* and *D Y o n i s i s*. We have applied this technique in the past to large vocabulary surname recognition as well (9). The resulting WFSA is depicted in Figure 1a. Transitions from one source state to all its possible destinations are considered to have equal probabilities. That is, transitions from the initial state, labelled as *Y* and *D*, have both a probability of 0.5. Alternatively the probabilities can be estimated by using training corpora. In order to build the input for our algorithm and construct the corresponding WFST we perform an alignment between the graphemic and phonetic forms of the words taking into account the correspondence between graphemes and phonemes, e.g., δ to *D*, ν to *n*, etc. The results are:

Y:γ i:ι a:ά n:ν ∈:ν i:η s:ς

Y:γ ∈:ι a:ά n:ν ∈:ν i:η s:ς

D:δ i:ι o:ο n:ν i:ύ s:σ i:η s:ς

(a)



(b)

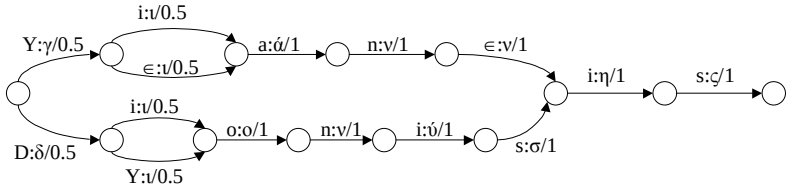


Figure 1. (a) Lexicon formed as WFSA, (b) Lexicon formed as WFST.

D:δ Y:ι o:ο n:ν i:ύ s:σ i:η s:ς

These strings constitute the input to our algorithm and the extracted WFST is given in Figure 1b. The empty symbol is denoted as ϵ. The transition probabilities are estimated in the same way as for WFSAs. This transducer is convenient for speech recognition because it accepts phonemes as input and produces words in graphemic form as output. If the input and output symbols are interchanged then the transducer becomes suitable for speech synthesis because it accepts letters as input and gives phonemes as output.

LANGUAGE MODEL

A language model can also be represented both as a WFSA and a WFST. An example is given in Figure 2, which depicts a simple language model at the dialogue state where the user gives the name of the person s/he would like to speak to. Possible speaker utterances (Greek form and English translation) that can be predicted by the language model are:

θέλω να μιλήσω με το {γιάννη, διονύση}
I want to speak with {john, dennis}

θα ήθελα να μιλήσω με το {γιάννη, διονύση}
I would like to speak with {john, dennis}

Transition probabilities are estimated by using training corpora. Figure 2b represents the same language model as Figure 2a by giving each transition identical input and output labels. This adds no new information, but it is a convenient way of interpreting any WFSA as WFST.

SYNTACTIC STRUCTURE

The transducer of Figure 3a gives the POS of the words contained in the language model of Figure 2. That is, it provides a link between the graphemic form of the word and the part-of-speech tag associated with it. Transition probabilities are not considered. Figure 3b shows a FST, which represents the rules of a context-free grammar. These rules are:

$V \rightarrow VP$ $\quad\quad$ $VP \rightarrow S$

$C\ V \rightarrow VP$ $\quad\quad$ $VP\ PP \rightarrow S$

$P\ NP \rightarrow PP$ $\quad\quad$ $VP\ VP \rightarrow S$

$A\ N \rightarrow NP$ $\quad\quad$ $VP\ VP\ PP \rightarrow S$

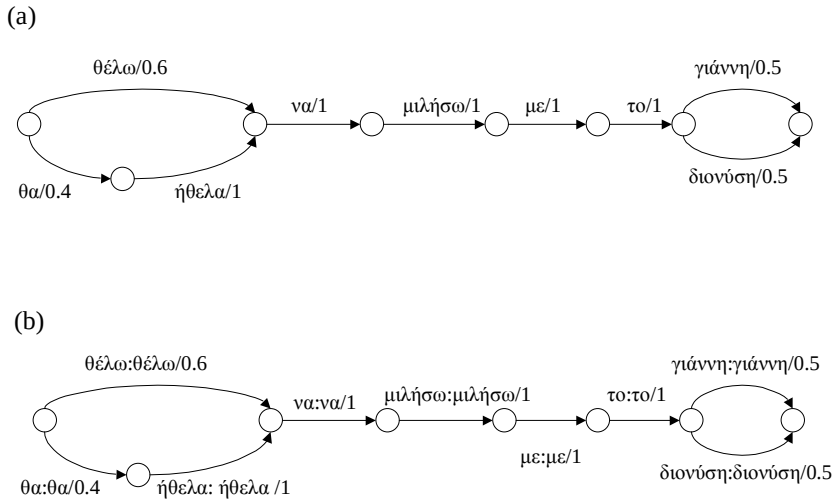


Figure 2. (a) Language model formed as WFSA, (b) Language model formed as WFST.

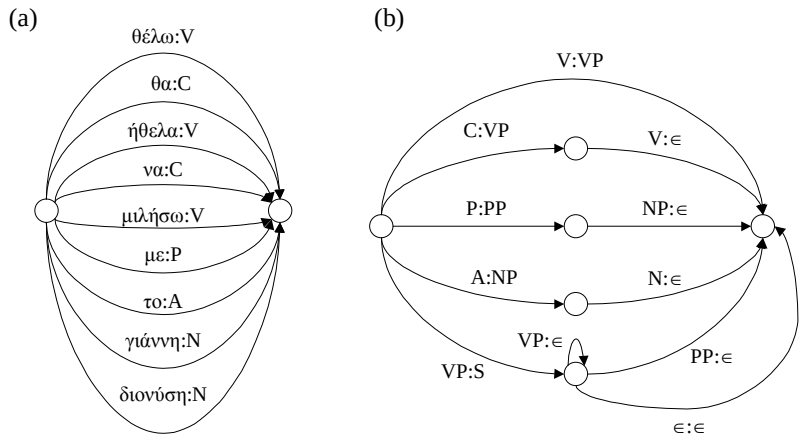


Figure 3. (a) Word-to-POS formed as FST, (b) FST that describes context-free grammatical rules.

where *S*, *NP*, *VP* and *PP* stand for sentence, noun phrase, verb phrase and prepositional phrase respectively, and *A*, *N*, *V*, *C* and *P* for article, noun, verb, conjunction and preposition.

The transducers of Figure 3 are convenient for parsing because if they are combined (see next section) they accept word sequences as input and produce syntactic structures as output. If the input and output symbols are interchanged they become suitable for natural language generation because they accept syntactic structures as input and give sentences as output.

TRANSDUCER COMPOSITION

Conventional speech recognisers are based on substitution (10). That is, each word of the language model is replaced by its phonetic transcription contained in the lexicon. Monophones are expanded to context-dependent triphones and there is also cross-word context expansion. Then each context-dependent triphone is substituted by its corresponding HMM. In the same way, a transducer composition algorithm is used to combine the various modelling transducers: acoustic models, pronunciation lexicon and language model.

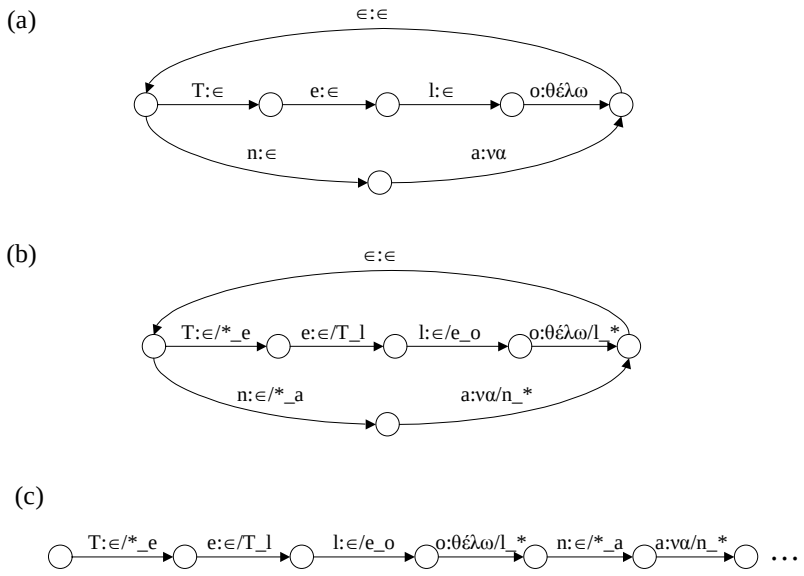


Figure 4. (a) Lexicon formed as FST, (b) lexicon considering context-dependency formed as FST, (c) composition of lexicon and language model transducers.

The composition of two weighted transducers *S* and *T* is a transducer *S* o *T* that assigns the weight *w* to mapping from symbol sequence *x* to sequence *z* just in case there is some symbol sequence *y* such that *S* maps *x* to *y* with weight *u*, *T* maps *y* to *z* with weight *v* and *w* = *u* + *v*. The states of *S* o *T* are pairs of a state of *S* and a state of *T*, and the arcs are built from pairs of arcs from *S* and *T* with paired source and destination states such that the output of the *S* arc matches the input of the *T* arc (3).

Figure 4a shows a lexicon formed as a FST, which contains the words *θέλω* and *να* of the language model depicted in Figure 2. An ϵ-transition is connected from the final state to the initial state. The resulting pronunciation lexicon *L* would pair any sequence of words from that vocabulary to their corresponding pronunciations. Thus *L* o *G* gives a transducer that maps from phones to word sequences restricted to the language model *G* and if it is composed with transducers that represent the acoustic models, it will form a WFST suitable for speech recognition, similar to the expanded network used in substitution-based recognisers. Note that the form of the lexicon has changed. That is, all transitions are labelled with the empty output symbol apart from the last one, which is labelled by the word in its graphemic form. This is necessary so that the lexicon and language model transducers can be combined.

The lexicon transducer of Figure 4b is the same as the one in Figure 4a, but additionally context-dependency has been considered. Cross-word dependencies may also be taken into account. In this way, FSTs can prove suitable for directly encoding context-dependent grapheme-to-phoneme rules. If we had a dictionary lookup instead it would be difficult to provide pronunciations at word junctures, where many phonological phenomena involve phoneme insertions or deletions. Most of the important ones obey simple contextual regularities, and it would be interesting to capture them in rules rather than treat them as additional statistical variability to be accounted for by the acoustic HMMs. Taking into consideration such phenomena can significantly reduce error rate. However, introducing cross-word dependencies in a conventional recogniser requires substantial changes in the search algorithm, or in the network expansion problem (11). In contrast, introducing cross-word phonological rules in a transduction simply means using a more complex FST to encode grapheme-to-phoneme rules rather than simple lookup (1).

Figure 4c depicts the composition of the lexicon transducer of Figure 4 and the language model transducer of Figure 2. Due to space limitations only a part of the resulting transducer is given. If the input and output symbols are interchanged then the transducer becomes suitable for speech synthesis. As it was mentioned in the previous section, if the FSTs of Figure 3 are combined they will produce a transducer that can be used for syntactic parsing and in the opposite direction for natural language generation. Our future work will focus on developing an algorithm for

composing WFSTs and in the sequel for performing search on transducers. This will prove the efficiency of the proposed model in practice.

CONCLUSIONS

In this paper we presented a transducer-based model for speech and language processing, which consists of a serious of layers of interconnected WFSTs. In order to create the transducers we used an algorithm that was first developed for constructing acyclic FSAs and then it was updated and improved to handle WFSTs. This algorithm supports real-time incremental construction of WFSTs. The structure of the different layers was explained in detail. That is, it was described how WFSTs can formulate lexicons, language models and rules of context-free grammars. Moreover, the composition of WFSTs was discussed so that the resulting transducer can be used for speech recognition and synthesis, or for syntactic parsing and natural language generation. The implications on the lexicon form were also presented. Future work will focus on implementing an algorithm for composing transducers and perform search on them. This will prove the efficiency of the proposed model in practice.

REFERENCES

1. Boulianne, G., Brousseau, J., Ouellet, P., Dumouchel, P., *French large vocabulary recognition with cross-word phonology transducers*, in Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 2000), Istanbul, Turkey.
2. Mohri, M., *Finite-state transducers in language and speech processing*, Computational Linguistics, Vol. 23, No. 2, pp. 269-312, June 1997.
3. Mohri, M., Riley, M., Hindle, D., Ljolje, A., Pereira, F., *Full expansion of context-dependent networks in large vocabulary speech recognition*, in Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'98), Seattle, WA.
4. Mohri, M., Riley, M., *Integrated context-dependent networks in very large vocabulary speech recognition*, in Proceedings of the 6th European Conference on Speech Communication and Technology (Eurospeech'99), Hungary, Budapest.
5. Mohri, M., Pereira, F., Riley, M., *Weighted finite-state transducers in speech recognition*, in Proceedings of ASR2000, International Workshop on Automatic Speech Recognition: Challenges for the Next Millennium, Paris, France, September 18-20, 2000.
6. Kuich, W., Salomaa, S., *Semirings, Automata, Languages*, Number 5 in EATCS Monographs on Theoretical Computer Science, Springer-Verlag, Berlin, Germany, 1986.
7. Sgarbas, K., Fakotakis, N., Kokkinakis, G., *Two algorithms for incremental construction of directed acyclic word graphs*, International Journal on Artificial Intelligence Tools, World Scientific, 4(3):369-381, 1995.
8. Sgarbas, K., Fakotakis, N., Kokkinakis, G., *Incremental construction of compact acyclic NFAs*, in Proceedings of ACL 2001, Toulouse, France.
9. Georgila, K., Sgarbas, K., Fakotakis, N., Kokkinakis, G., *Fast very large vocabulary recognition based on compact DAWG-structured language models*, in Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP 2000), Vol. 2, pp. 987-990, Beijing, China.
10. Valtchev, V., Odell, J., Woodland, P., Young, S., *A dynamic network decoder for large vocabulary speech recognition*, in Proceedings of the 3th International Conference on Spoken Language Processing (ICSLP'94).
11. Beyerlein, P., Ullrich, M., Wilcox, P., *Modelling and decoding of crossword context dependent phones in the Philips large vocabulary continuous speech recognition system*, in Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech'97), Vol. 3, pp. 1163-1166, Rhodes, Greece.