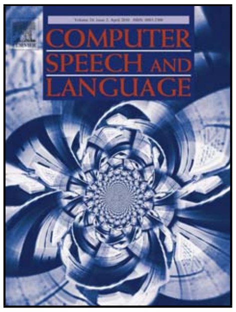
Semantic and Phonetic Automatic Reconstruction of Medical Dictations

Stefan Petrik, Christina Drexel, Leo Fessler, Jeremy Jancsary, Alexandra Klein, Gernot Kubin, Johannes Matiasek, Franz Pernkopf, Harald Trost

Accepted Manuscript



Title: Semantic and Phonetic Automatic Reconstruction of Medical Dictations

Authors: Stefan Petrik, Christina Drexel, Leo Fessler, Jeremy Jancsary, Alexandra Klein, Gernot Kubin, Johannes Matiasek, Franz Pernkopf, Harald Trost

PII: S0885-2308(10)00058-6
DOI: doi:10.1016/j.csl.2010.07.003
Reference: YCSLA 468

To appear in:

Received date: 14-8-2008
Revised date: 5-7-2010
Accepted date: 6-7-2010

Semantic and Phonetic Automatic Reconstruction of Medical Dictations

Stefan Petrik[a,*] Christina Drexel[d] Leo Fessler[d]
Jeremy Jancsary[b] Alexandra Klein[b] Gernot Kubin[a]
Johannes Matiasek[b] Franz Pernkopf[a] Harald Trost[c]

[a]*Signal Processing & Speech Communication Laboratory, Graz University of Technology, Graz, Austria*

[b]*Austrian Research Institute for Artificial Intelligence, Vienna, Austria*

[c]*Institute of Medical Cybernetics and Artificial Intelligence of the Center for Brain Research, Medical University Vienna, Austria*

[d]*Nuance Communications Austria, Vienna, Austria*

Abstract

Automatic speech recognition (ASR) has become a valuable tool in large document production environments like medical dictation. While manual post-processing is still needed for correcting speech recognition errors and for creating documents which adhere to various stylistic and formatting conventions, a large part of the document production process is carried out by the ASR system. For improving the quality of the system output, knowledge about the multi-layered relationship between the dictated texts and the final documents is required. Thus, typical speech-recognition errors can be avoided, and proper style and formatting can be anticipated in the ASR part of the document production process. Yet – while vast amounts of recognition results and manually edited final reports are constantly being produced – the error-free literal transcripts of the actually dictated texts are a scarce and costly resource because they have to be created by manually transcribing the audio files.

To obtain large corpora of literal transcripts for medical dictation, we propose a method for automatically reconstructing them from draft speech-recognition transcripts plus the corresponding final medical reports. The main innovative aspect of our method is the combination of two independent knowledge sources: phonetic information for the identification of speech-recognition errors and semantic information for detecting post-editing concerning format and style. Speech recognition results and final reports are first aligned, then properly matched based on semantic and phonetic similarity, and finally categorised and selectively combined into a reconstruction hypothesis. This method can be used for various applications in language technology, e.g., adaptation for ASR, document production, or generally for the development of parallel text corpora of non-literal text resources. In an experimental evaluation, which also includes an assessment of the quality of the reconstructed transcripts compared to manual transcriptions, the described method results in a relative word error rate reduction of 7.74% after retraining the standard language model with reconstructed transcripts.

Key words: automatic transcription, semantics, phonetics, automatic speech recognition, dictation

1 Introduction

After decades of research, speech recognition technology has reached a level where it can be successfully integrated into products for everyday use. In particular, this applies to dictation systems with integrated speech recognition which help reduce the amount of manual transcriptions. In the medical domain, where dictation traditionally plays an important role, speech recognition systems have contributed to a more efficient report creation process since medical transcriptionists no longer have to type whole documents, instead they only do the post-processing to create the final reports. This way, highly skilled medical transcriptionists make better use of their expertise.

In many cases, this post-processing step unfortunately still involves a lot of tedious editing: recognition errors have to be corrected, and the style and formatting of the document have to be adapted to the standards applied to written reports. Particularly for dictations by unexperienced users, post-processing can become time consuming, and thus may lead to many and various deviations between the recognition results and the final reports.

While recognition results and final reports are usually available in abundance, manual transcriptions of the actual spoken words without recognition errors (i.e., assuming perfect recognition) are costly and scarce. For training automatic speech recognition systems, however, literal transcriptions of the actual words are needed.

A standard methodology to overcome the problem of non-literal transcriptions in ASR training is unsupervised or lightly supervised training (Lamel et. al. (2002), Kemp and Waibel (1999)). These approaches allow the generation of statistical models from only small amounts of literal transcriptions together with large amounts of non-literal transcriptions in an iterative fashion. For language model training, methods like linear model interpolation

* Corresponding author.
Email address: stefan.petrik@tugraz.at (Stefan Petrik).

(Chen and Goodman (1996)) or transformation-based learning (Peters and Drexel (2004)) are used to cope with non-literal transcriptions. Although these methodologies lead to reductions in word error rate for as various domains as news broadcasts or transcriptions of class lectures (Hazen (2006)), they do not give explanations for the mismatches between the non-literal data and the actual wording in the training utterances. Furthermore, problematic segments like disfluencies, hesitations, or speaker corrections cannot be modelled without proper annotation. For these reasons, literal transcriptions are still valuable.

These motivations lead us to the definition of the problem of how a literal transcription can be automatically reconstructed from non-literal transcripts of different information sources. This problem has already been addressed by Pakhomov et al. (2001) for modelling disfluencies and hesitations in medical dictations. However, a comprehensive model for automatic reconstruction needs to go beyond the scope of specific phenomena and provide a generic framework for exploiting the full potential of the analysed documents.

In this article, we propose such a reconstruction framework and describe a system which has been developed for automatically reconstructing the actual spoken words from the recognition result and the final medical reports. These two different input sources are complementary for the task of reconstructing literal transcripts. The resulting reconstructions can be used the same way as manual transcriptions for training speech recognition systems.

The base for reconstruction is an alignment between the written report and the recognition result. The alignment takes into account semantic information (for explaining reformulations) and phonetic information (for explaining recognition errors) as well as syntactic information in terms of document formatting. From the interpretation of the deviations between the written report and the recognition result, the words which are considered to have actually been spoken are reconstructed. According to the proposed methods, we name our approach Semantic and Phonetic Automatic ReConstruction (SPARC). The main innovative aspect of our method is the optimal interplay between two independent knowledge sources, namely semantics and acoustics/phonetics in the categorisation of differences between automatic transcript and final document, as well as in the reconstruction of the original utterance from these two data sources.

Qualitative and quantitative evaluations based on manual transcriptions have shown that, in many cases, the alignment leads to a correct reconstruction. The resulting reconstructed text can serve not only as a base for training and improving the speech recognition system; a deeper understanding of the typical reformulations and reformatting may eventually also support a shift from mere speech recognition to document production in dictation applications.

In the following sections, we will continue with a more detailed account of the SPARC approach in section 2 and a description of the available text corpora in section 3. Following this introductory part, in sections 4, 5, and 6 the three main units of the approach - text alignment, similarity measurement, and text reconstruction - are then elucidated. In section 7, we report experimental results in terms of the quality of the reconstructed text and an automatic speech recognition experiment with retrained language model. We conclude the paper with a discussion of the results and an outlook for further applications.

2 SPARC approach

The SPARC approach is a method for the automatic production of literal transcriptions from available data sources in large document production environments using speech recognition. Three types of data are currently available in such systems:

- *Audio files (AF)*, comprising the original utterances;
- *Draft transcriptions (DT)* - or more simply: *recognised texts* -, produced by the dictation system (containing the recognition errors);
- *Final documents (FD)* - or more simply: *written texts* -, produced by the typist (where recognition errors are corrected but where also some parts are re-formulated in a way different from the original utterances).

Error-free *literal transcriptions (LT)* - or more simply: *reference texts* - of the audio files, however, are usually not available, or only to a certain degree if some manual transcriptions have been made. Yet, literal transcriptions of the original spoken utterances are needed for advancing the accuracy and efficiency of automated dictation:

- Aligned corpora of LT and FD can be used to automatically learn recurrent reformulations, thus allowing automated dictation to be augmented by an automatic text reformulation module which provides a draft that is closer to the intended final document.
- Large quantities of literal transcriptions and audio files can serve as data for training of the acoustic and language models to decrease the word error rate of speech recognition.

For medical dictations, the reconstruction task was already described by Pakhomov et al. (2001). There, the authors propose an augmented probabilistic finite-state model for generating semi-literal transcriptions. This probabilistic model handles so-called ‘out-of-transcription expressions’ like greetings, false starts and repairs, and filled pauses as the only sources of mismatches between recognised and written texts. For the same task, SPARC provides added value by also explaining and categorising such mismatches. For hypothesising the reconstructed text mismatches are not only detected, but also interpreted. The interpretation of a mismatching token pair as e.g., a recognition error, or a reformulation of the typist helps in designing more accurate models for the differences between spoken and written form of medical dictations. Voll (2006) describes a hybrid method for detecting speech recognition errors in radiology reports based on semantic knowledge, constraint rules and statistical modelling (i.e., pointwise mutual information and co-occurrence analysis). In Hazen (2006), transcription generation was presented for recorded academic lectures with a finite-state transducer approach.

Semantic relatedness and similarity measures have mostly been developed to improve the recall of Information Retrieval (IR) systems. There are two main established ways of measuring the semantic similarity between two terms: on the one hand, relatedness can be measured in terms of the distance between two words or multiword expressions in a knowledge base, e.g., WordNet (see Fellbaum (1998)). On the other hand, relatedness can be derived from a corpus by determining co-occurrence and context features with IR methods. Often, corpus- and knowledge-based measures are combined. Due to the many available knowledge sources, the medical domain lends itself well to knowledge-based measures for semantic relatedness and similarity (for an overview, cf. Pedersen et al. (2007)).

Similarly, phonetic similarity measurement has been used for addressing many topics in ASR: modelling pronunciation variation (e.g., Ristad and Yianilos (1998), Filali and Bilmes (2005)), predicting ASR errors (Fosler-Lussier et al. (2005)), measuring acoustic confusability (Printz and Olsen (2002)), discriminative language model training and OOV detection (Rastrow et al. (2009)), or IR (Zobel and Dart (1996)). In many of these applications, confusion matrices are used to measure the phonetic similarity of phone sequences or phone confusion networks. These matrices are either handcrafted, e.g., from phonetic class information, or estimated from data.

The technological goal is to automatically construct an error-free literal written transcription of the user's original utterances. Methodologically, the basis for this reconstruction is formed by an analysis of the semantic and acoustic differences between DT and FD. Scientifically, SPARC requires solutions for the following problems:

- Automatic semantic annotation of text corpora with the help of a domain-specific ontology.
- Accurate text alignment and chunking for the available draft transcriptions and final documents.
- Methods for comparing aligned text chunks for semantic and phonetic similarity.

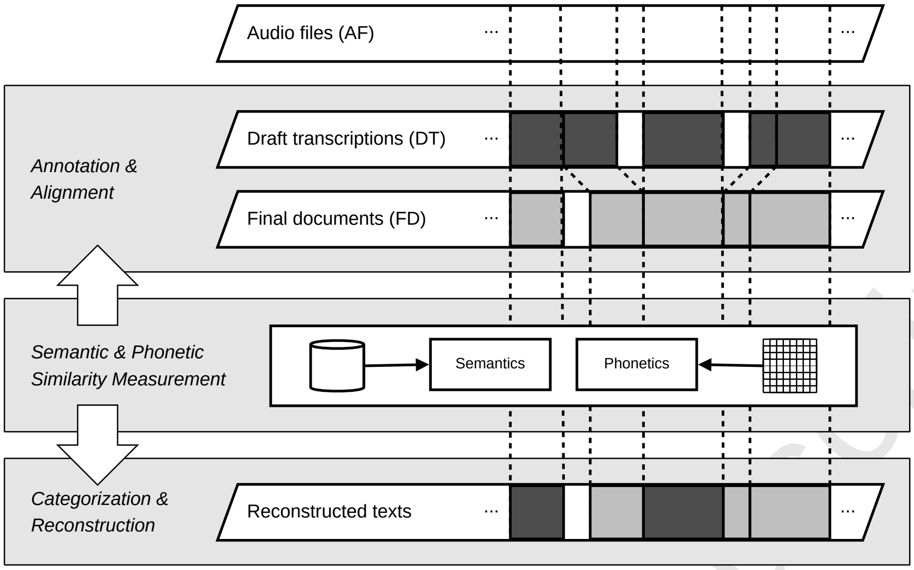

Audio files (AF)
Annotation & Alignment
Draft transcriptions (DT)
Final documents (FD)
Semantic & Phonetic Similarity Measurement
Semantics
Phonetics
Categorization & Reconstruction
Reconstructed texts

Fig. 1. SPARC architecture: draft transcriptions and final documents are first annotated, then properly matched based on semantic and phonetic similarities, and finally categorised and selectively combined into a reconstruction hypothesis.

- Classification of text chunks based on the similarity measures (text reconstruction).

Figure 1 illustrates the architecture of the SPARC approach. Our method starts with the automatic semantic annotation of both DTs and FDs. Pairs of documents are then aligned to identify chunks where texts display differences. Semantic similarity is measured based on the semantic annotation, while phonetic similarity is determined online with a parameterised stochastic similarity measure. This way, the difference between a specific chunk in DT and FD can be categorised as correction of a speech recognition error or a reformulation by the human typist – or a combination of both (cf. table 1). Reconstruction of the originally dictated words is based on this analysis. Note that semantic and phonetic similarity measurement are used for both alignment and reconstruction.

SPARC can be adapted to any domain and to any language as long as the basis for training/learning – namely adequately sized parallel corpora of DT and FD – as well as the necessary linguistic resources – lexical, morphology, thesauri, etc. – are available. We implemented SPARC for English medical reporting, due to the fact that very large collections of medical corpora in English can be obtained, and medical reporting is at the moment by far the most important application of speech recognition in professional dictation.

		PHONETICS	
		similar	*dissimilar*
SEMANTICS	*similar*	MATCH	REFORMULATION
	dissimilar	CORRECTION	REFORMULATION & CORRECTION

Table 1
The SPARC approach to text reconstruction. Based on semantic and phonetic similarity measurements, chunks of written and recognised text can be classified as either matches, reformulations, corrections or a combination of both.

3 Data Description

For reconstruction, we distinguish between matching (i.e., identical) and mismatching parts of the aligned texts. As this task is trivial for matching parts, only the mismatching parts will be of interest. Generally, we describe mismatches between texts on word level in terms of the mismatch edit operations insertion, deletion, and substitution. This way, a word error rate can be determined easily, but mismatch interpretation is difficult since actual mismatches can be composed of several adjacent mismatch edit operations. For this reason, we define a *mismatch region* as a contiguous sequence of mismatch edit operations in order to establish correspondences between matched words.

A statistical study of a corpus of 80,000 medical reports with 38 million words revealed an average length of 2.3 words for a mismatch region and an average occurrence of 3.6 times for this region within the corpus. Regions occurring only once account already for 60% of all mismatches while frequent regions occurring $\geq$ 10,000 times only account for about 11% of all mismatches. Such highly frequent mismatches are, e.g., insertions or deletions of punctuations and short words. On the other hand, regions of length 1 cover around 20% of all mismatches, and 75% of all mismatches occur in regions of length $\leq$ 5. For the reconstruction task, this means that only relatively short symbol sequences have to be processed.

Mismatches can be traced back to the human dictation process, the automatic recognition process, and the human transcription process. In general, the dictating person speaks freely, thus hesitations, self-corrections, and repetitions can be observed often in the recordings, but of course not in the final documents. ASR is error-prone, resulting in the confusion of words which are phonetically similar. The transcription process completes the range of mismatch sources by adding formatting to the text according to certain well-defined standards. Formatting affects the text in two ways: first, by additional structure like inserted punctuations, paragraph breaks, or capitalisation of words, and second, by formatting of particular document entities like headings, grammatical units (dates, quantities, etc.), or enumerations out of continuous text. The latter formatting step makes reconstruction difficult, as different speaking variants are mapped onto a standardised written form. Furthermore, the structure and style of the text can be altered by reformulations of the typist as well. These alterations include expansion of abbreviations, acronyms, and short forms, or grammatical corrections like changes in genus, tempus, or numerus so as to put the final written text into a proper stylistic and grammatical form.

4 Text alignment

Establishing proper alignment of the final report (FD) and the recognised text (DT) is an important prerequisite for all further steps (Huber (2006)).

During alignment, both input documents are viewed as sequences of tokens. A generalised Levenshtein alignment algorithm is then applied to these sequences (cf. Levenshtein (1966)). The Levenshtein algorithm views alignment as a minimisation problem, where a number of actions with associated costs can be performed to navigate through the search space:

- If **substitution** is performed for two elements x_i and y_j of sequences x_1^N and y_1^M, then these two elements will be mapped to each other in the final alignment and labelled with [=]. This action includes the special case of identity where $x_i = y_j$ with zero cost (unlike 'true' substitutions).
- **Deletion**, on the other hand, results in element x_i of sequence x_1^N being mapped to the empty element, i.e., it will not have a corresponding element of sequence y_1^M in the alignment. Deletions are labelled with [<].
- **Insertion** is symmetric to deletion and as such leads to y_j being mapped to the empty element. Insertions are labelled with [>].

For each pairing (x_i, y_j) out of $x_1^N \times y_1^M$, a scoring function is invoked that evaluates the respective costs for each of the three available actions. Dynamic programming is applied to find the cheapest path (i.e., the cheapest sequence of actions) through the search space in $O(NM)$ time, where N and M are the length of x_1^N and y_1^M, respectively. This approach allows to factor out all domain-specific aspects to the scoring function by, e.g., assigning special scores to formatting marks while the dynamic programming scheme for cost minimisation remains untouched.

A common phenomenon that can be observed in such alignments are mismatches caused by recognition errors involving splitting or merging of words

written text	↔	*recognised text*
.	COR	.
\n\n/DIET/ \n	=	\n\n
Low-fat	=	Diet
,	=	is
low-cholesterol	=	a
	>	low
	>	fat
,	COR	,
two-gram	=	low
	>	cholesterol
	>	2 grams
sodium	COR	sodium

written text	↔	*recognised text*
.	COR	.
	>	\n \n
\n\n/DIET/ \n	=	Diet
	>	is
	>	a
Low-fat	=<	low
	<=	fat
,	=	,
low-cholesterol	=<	low
	<=	cholesterol
,	<	
two-gram	=	2 grams
sodium	COR	sodium

Fig. 2. A sample text passage with mismatch regions highlighted in boxes, aligned with standard Levenshtein alignment (left) and the advanced multi-alignment computed by SPARC (right). Labels are: COR for identical words, [=] for corresponding/substituted words, [<] for deletions, and [>] for insertions.

(segmentation errors) within the recognised texts or massive reductions due to fast speech (cf. fig. 2 and 4). To account for these problems, the alignment has been extended to handle multiple levels of segmentation. Since the alignment procedure operates on sequences of tokens, it can be applied recursively to any pair of tokens that has been further split to a finer level of segmentation. Multi-word expressions or grammatical units can thus be reduced to sequences of single words which in turn can be broken down to sequences of syllables. The sequence of alignment labels obtained from these alignment processes are concatenated into a single alignment label, expressing the amount of overlap on submatching level between the parent tokens.

For the purpose of creating a literal transcript, it is crucial that all *corresponding* passages of the two input documents are mapped to each other. *Corresponding* means that two passages denote the same section in the actual dictation. Naturally, the two passages need not necessarily consist of the same tokens like, e.g., in a mismatch region. Figure 2 illustrates this problem for a sample text passage. The standard Levenshtein algorithm with equal costs for all edit operations calculates the minimum cost alignment based on the orthographic spelling, however, at the expense of proper word correspondences. Furthermore, the mismatch region is even split improperly at the wrong comma, such that the semantic correspondence between written and recognised text is lost. The SPARC alignment re-establishes the proper word correspondences even for segmentation mismatches and preserves a singular contiguous mismatch region.

Hence, two scoring mechanisms have been developed that compare token pairs for semantic and for phonetic similarity, respectively, and these have then been united in a single scoring function. Naturally, it would be desirable to not only compare token pairs, but whole passages for similarity in the scoring functions. However, the restriction to token pairs is a necessary concession to the already unfavourable computational complexity of alignment problems. Less local comparisons can be performed at the reconstruction stage.

5 Similarity measures

Similarity measurement of tokens is used in both text alignment (cf. section 4) and reconstruction (cf. section 6). For text alignment, the similarity measures are consulted by scoring functions of the generalised Levenshtein alignment algorithm to improve accuracy in contrast to plain orthographic matching. In text reconstruction, the measures are used to condition reconstruction rules and perform the classification of text chunks as either matches, corrections, reformulations, or a combination thereof (cf. section 2, table 1). The basic methods, however, are the same in alignment and in reconstruction.

5.1 Semantic similarity

In order to measure semantic similarity, words are first assigned a semantic representation. Since our primary application domain is medical reports, specialised medical terminology has to be incorporated into the knowledge sources. The resource we employ for that purpose is the Unified Medical Language System (UMLS, Lindberg et al. (1993)), which includes a metathesaurus, a semantic network, and a lexicon (SPECIALIST). The morphosyntactic information from the lexicon was worked into the finite-state transducer that is used as a morphological lexicon.

The metathesaurus is a very large, multi-purpose, and multi-lingual terminology database that contains information about biomedical and health related concepts, their various names, and the relationships among them. Unfortunately, the relations between UMLS concepts appear to depend on the particular knowledge source the concept comes from, and the depth it is modeled solely within that knowledge source. Nevertheless, for analysing synonymity of two words or determining a rough degree of semantic relatedness, these relations appear to be sufficient. In addition, all concepts in the metathesaurus are assigned to at least one semantic type from the UMLS semantic network.

Furthermore, a high coverage resource for general vocabulary, the WordNet lexical database (Fellbaum (1998)) is available for English. In Wordnet, nouns, verbs, adjectives and adverbs are organised into synonym sets, each representing one underlying lexical concept; the relations connecting WordNet synsets are quite different from the relations between UMLS concepts. For our purpose, the hypernym relation is the most important synset relation. [1].

The following ordinal scale has been defined in order to obtain a rough measure of semantic similarity of two words:

7	identical (modulo case)
6	same root (only inflection)
5	synonymous
4	morphologically derived
3	conceptual siblings
2	same UMLS semantic type or `parent(word1,word2)` or `parent(word2,word1)`
1	direct hierarchical relation between semantic types
0	no similarity at all

In the above context, `parent(word1, word2)` means that `word1` maps to a concept/synset (inter alia) that is a direct UMLS superconcept or hypernym synset of one of the concepts/synsets `word2` maps to. Two words are siblings if they share at least one direct UMLS superconcept or hypernym synset. The intuition behind this was to use a measure which is available in both WordNet and UMLS, which has a finer granularity than the (rather crude) UMLS semantic type and which assures that both concepts have something in common (the “supertype”).

Based on the similarity value of its two argument tokens on the ordinal scale, costs for substitution, insertion and deletion are determined by the semantic scoring function and returned to the invoking alignment framework.

5.2 Phonetic similarity

Phonetic similarity measurement (Petrik and Kubin (2007)) requires three sources of information for comparison: the phonetic symbol sequence from the recognised text, the orthographic word sequence from the recognised text and the word sequence from the written text. The basic similarity measurement

[1] For a study which compares WordNet and UMLS in greater detail, see Burgun and Bodenreider (2001)

process is depicted in figure 3, and its main components are explained in more detail below.

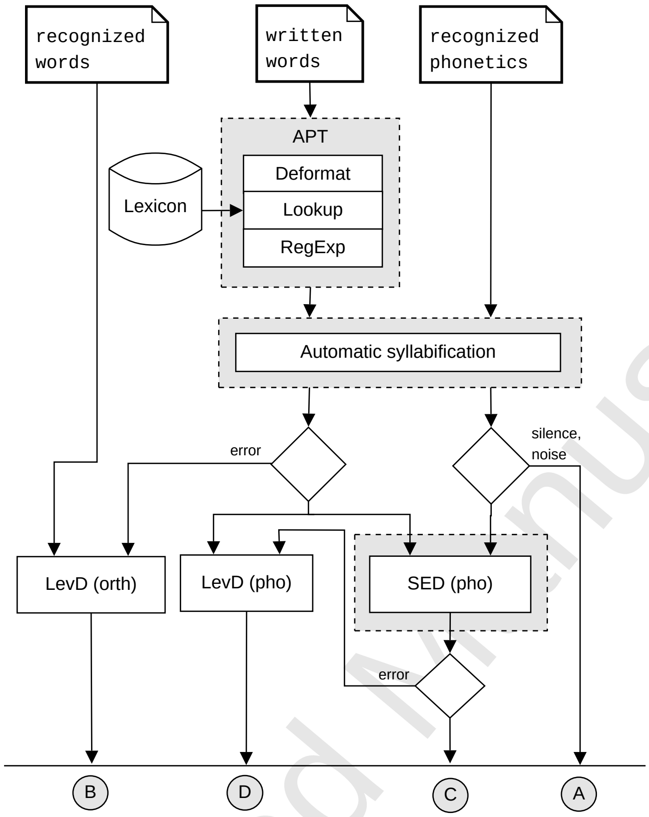


Fig. 3. Block scheme of phonetic similarity function: automatic phonetic transcription (APT), automatic syllabification, trainable string edit distance measure (SED), and Levenshtein measure (LevD)

5.2.1 Automatic phonetic transcription (APT)

In a first step, the written text is transferred to the phonetic domain with automatic phonetic transcription (APT). This is done by a simple lexicon lookup. The phonetic lexicon we used contains 160,000 words with 197,000 pronunciations. It includes common as well as domain-dependent vocabulary and was compiled from customary and publicly available resources like CMU-dict [2] . To improve coverage on formatted text parts, a de-formatting grammar is applied to formatted text units. The de-formatting grammar is an inverted version of a formatting grammar used in the speech recogniser which now produces speaking variants for a given formatted entity as shown in the following example:

[2] See `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`

```
December 6  →  December the sixth
               December O six
               sixth of December
               ...
```

Furthermore, a simple regular expression syntax was defined to encode the possibly many speaking and pronunciation variants in a single string. The extended syntax allows grouping and alternation ("|") of expressions as described in the corresponding BNF grammar:

$$\begin{aligned} \mathit{expr} &:= \mathit{group}^{+} \\ \mathit{group} &:= \text{“(”}\ \mathit{word}^{+}\ (\ \text{“|”}\ \mathit{word}^{*})^{*}\ \text{“)”} \\ \mathit{word} &:= [A, .., Z, a, .., z] \end{aligned}$$

Since the word after the alternation-operator | is optional, whole words may be omitted. This is particularly useful for dealing with hesitations or dictated formatting instructions which do not appear in the written text by definition.

The recognised text still contains non-speech events like silence or noise markers which do not have a phonetic transcription and which are not contained in the written text either. These parts get scores assigned which automatically force them to be marked as insertions (path A in figure 3). After that, it is certain that the remaining string pairs are valid phonetic strings that can be handled by the phonetic similarity measurement model. Whenever the APT fails, phonetic matching is impossible, so the string pair can only be matched in the orthographic domain with the Levenshtein measure (path B in figure 3).

5.2.2 Automatic syllabification

Syllable boundaries are usually best assigned by expert phoneticians or can be inferred from stress markers stored offline in the lexicon. For the highly specific vocabulary used in the medical domain, such annotated expert phonetic lexica were not available to us. Furthermore, the vocabulary is subject to change over time, as new medication may be prescribed or medical treatments and measures may change. Therefore, an online automatic syllabification algorithm was implemented to determine syllables directly from the texts. The algorithm introduced by Hammond (1995) is based on Optimality Theory (Prince and Smolensky (2004)), where phonological processes are modelled by applying ranked constraints on base forms to obtain surface forms. For syllabification, this means that a number of competing syllabification constraints are applied to the input words. In contrast to Hammond (1995), the 'noonset' constraint had to be removed, as primary stress information was not available in the phonetic lexicon. The modified algorithm was tested on a sample set of 100 randomly selected words which were manually compared to a reference syllabification provided by Merriam-Webster's online dictionary[3]. The modification degraded the performance of the algorithm in terms of accuracy of the syllable boundaries, but not the number of detected syllables, and still returned correct results in around 80% of all cases.

With this algorithm, the word level units for recognised and written text are split into sequences of syllables. The alignment algorithm is then applied recursively on the syllable sequences. Adjacent words are not only aligned, but also tested for overlap on syllable level. The word-level alignment label is therefore replaced by an overlap symbol string. The resulting alignment expresses both word and syllable level correspondences. Consider the sample alignment in figure 4. Within the first mismatch region, the word `Charcot` was incorrectly recognised and split into `sharp` and `cold`. The syllable level alignment, however, shows that `sharp` corresponds to the first, and `cold` to the second syllable of `Charcot`. As syllable alignment is determined based on phonetic similarity, the alignment may sometimes look confusing. The short words `of` and `in` are not aligned with each other, since `in` is phonetically more similar to the last syllable of `ulceration` than to `of`.

5.2.3 Training a string edit distance measure (SED)

The main component of the phonetic scoring function is a trainable string edit distance measure based on the stochastic model presented by Ristad and Yianilos (1998). In this model, a string pair (x, y) is represented by all sequences of edit operations z_i which produce that pair. Assuming that each pair can be produced by at least one edit sequence, the probability of the pair is the sum of the probabilities of all edit sequences for that pair:

$$p(x, y|\theta) = \sum_{\{z^n\#: v(z^n\#)=\langle x,y\rangle\}} p(z^n\#|\theta) \; , \tag{1}$$

where # is the sequence termination symbol and $v(z^n\#)$ defines the set of all terminated edit sequences producing $\langle x, y\rangle$. Since every z_i has a probability $p(z_i)$ assigned and the model is memoryless, $p(z^n\#|\theta)$ is the product of the probabilities of the single edit operations. These probabilities $p(z_i)$ are learned from a corpus of predefined, similar string pairs with an EM algorithm (Ristad and Yianilos (1998)). Accumulating the probabilities for all edit sequences, a similarity measure can now be defined as

[3] See `http://www.merriam-webster.com/`

written text		↔	*recognised text*	
a		COR		a
Charcot	SAr·k@t	=<	SArp	sharp
--	--	>=	koUld	cold
foot	fUt	=	fUt	foot
,	kAm@	<	--	--
though	DoU	=	nO	no
there is	Der= Iz	COR	Der= Iz	there is
no	nO	COR	nO	no
ulceration	Vl·s@·reI·S@n	====	Ql·t@·reI·S@n	alteration
--	--	<<<=	In	in
of	Vv	<	--	--
skin	skIn	COR	skIn	skin

Fig. 4. A sample alignment containing two mismatch regions. The re-aligned mismatch regions are highlighted in boxes while identical words are labelled with COR. Phonetic strings are in SAMPA notation and syllable boundaries are marked with dots [·]. Note that the [=]-overlap symbol just indicates correspondence, not equality of syllables, in contrast to the insertion [<] and deletion [>] symbols which label non-matching syllables.

$$d(x, y) = -\log\ p(x, y|\theta)\ . \tag{2}$$

Two issues should be noted at this point. First, the similarity value decreases exponentially with the input string length due to the usage of the distinct termination symbol #. Therefore, the similarity value needs to be normalised – in this case by the sum of the input string lengths. Furthermore, the similarity measure is never zero since each edit operation has assigned a probability $0 < z_i < 1$. To still be able to detect exact matches, the systematic bias is subtracted symmetrically to normalise the measure to zero according to the following formula:

$$d_0(x, y) = d(x, y) - \frac{1}{2} \cdot [d(x, x) + d(y, y)] \tag{3}$$

Prior to matching, the regular expressions generated by the automatic phonetic transcription have to be expanded again, as only the minimum score for all possible realizations is returned (path C in figure 3). Finally, in case the stochastic model fails, another fallback to the Levenshtein measure is done, this time with phonetic strings (path D in figure 3).

The model was trained in 3 EM iterations with a set of 13,383 string pairs obtained from manual narrow phonetic transcriptions of a domain-specific corpus of 272 medical reports. The transcriptions were done by English students with specific training in phonetics, ensuring quality in the transcription process. For each word in the transcription, a string pair consisting of the canonical transcription obtained from the phonetic lexicon and the actual phonetic transcription was compiled. This way, phonetic similarity is clearly defined, and frequent phoneme confusions can be learned easily from real-world data.

Figure 5 displays the learned probability distribution for each edit operation defined on a phonetic symbol pair. As expected, most of the probability mass was assigned to identity operations (main diagonal). Furthermore, vowels were likely to be substituted by *schwa* (/@/) and vice versa. Voiced-unvoiced substitutions between `/t/` and `/d/` were also quite prominent, just like substitutions between the syllabic (`/n=/, /m=/, /l=/`) and non-syllabic forms (`/n/, /m/, /l/`) of the semi-vowels. The learned probability distribution clearly reflects the phonetic knowledge that can be observed in dictated speech.

5.3 Combined similarity measurement

The semantic and phonetic scoring functions are used as building blocks for a combined scoring function that best exhibits the behaviour that is required for further processing.

The goal is to align any two sequences of elements for which phonetic or semantic similarity can be assigned. Distinguishing between phonetic and semantic similarity is postponed to the reconstruction process since it is the single aim of this processing stage to put related elements into proper correspondences.

Combining the two sets of scores for substitution, deletion and insertion into a single set of scores is somewhat subtle, because contradictory actions might be suggested by semantic and phonetic similarity scores. As an example, phonetic scoring might vote for substituting two elements, while semantic scoring might want to substitute one of these elements with a different one. Such contradictions need to be resolved while still following the overall goal of performing substitution (mapping between two elements) when either the phonetic or semantic measure indicate similarity.

The combined scoring function for alignment was developed and tuned heuristically by manual inspection of a small number of alignments. In general, the phonetic similarity function analyses the tokens on a high level of detail and thus establishes correspondences in a greedy fashion which sometimes results

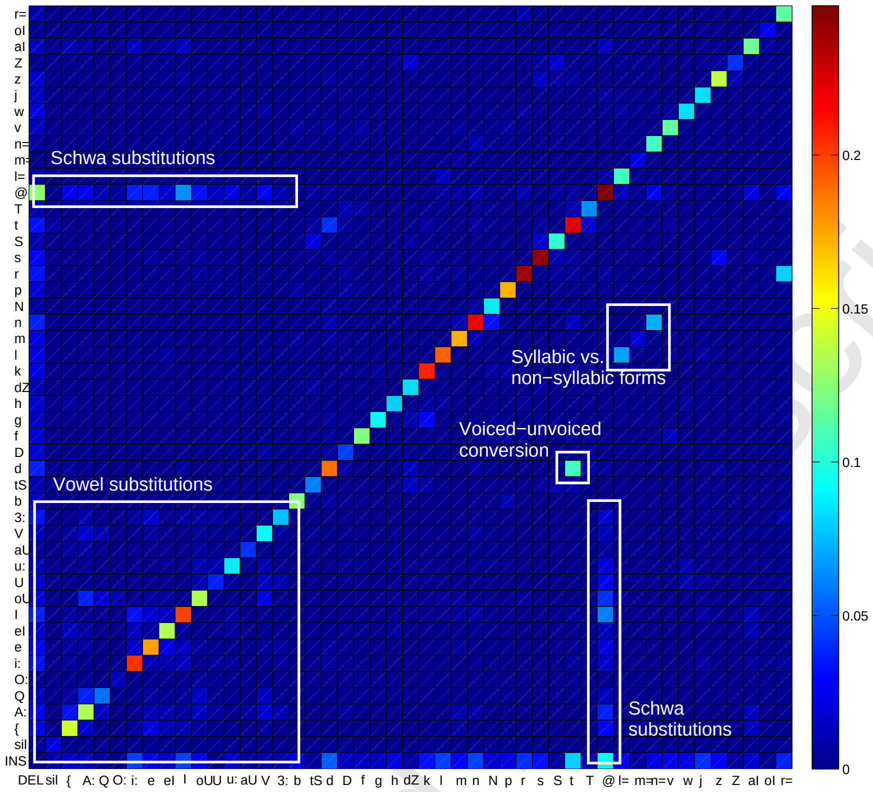


Fig. 5. Learned probability distribution for edit operations z_i after 3 EM iterations. Phonetic symbols are in SAMPA notation.

in alignments that cannot be interpreted meaningfully any more. The semantic similarity scoring function on the other hand, is more robust against "over-correspondencing" but at the same time not capable of properly detecting fine matches. For these reasons, semantic matching is applied in the first place to filter out clear cases and avoid overstretched regions of correspondence, before phonetic matching is used to find detailed matches.

6 Text reconstruction

Based on the alignment, a reconstruction hypothesis for a literal transcription can be computed. In general, this process can be seen as a classification task, as already outlined in section 2 (cf. table 1). A classifier is used to select the recognised or the written text for each alignment token. For optimal control and fine-tuning, we implemented a rule-based reconstruction system that allows generic and context-dependent analysis of the alignment. This approach

written text	↔	*recognised text*	*deformatted*	*reconstructed*	*rule*
He	=	he	he	he	sim_bigram
says	=	said	said	said	sim_bigram
he	COR	he	he	he	identical
did not	COR	did not	did not	did not	identical
have	COR	have	have	have	identical
any	COR	any	any	any	identical
cardiac	COR	cardiac	cardiac	cardiac	identical
	>	,	comma	–	–
	>	residual	residual	residual	repetition
residuals	COR	residuals	residuals	residuals	identical

Fig. 6. Reconstruction of a text passage with two mismatch regions (dashed boxes): Written text, alignment labels, and recognized text are given as input. Deformatted recognized text, reconstructed text, and the matching rule for each alignment line are generated by the system.

is also compared to state-of-the-art automatic classification approaches using the same input features.

The rule-based reconstruction process, which is described in Jancsary et al. (2007), operates on the established alignment. The steps performed for reconstructing the actually spoken words are the following (cf. figure 6):

- **Deformatting**:
  First, a column containing the completely deformatted variant of the recognised words is created (cf. section 5.2.1). In particular, formatted items and punctuation are replaced by the most likely spoken variant based on the phonetic representation and the measures for phonetic and semantic similarity.
- **Identifying and retracing moved blocks**:
  Then, moved blocks are identified if there are any, and within the written text the identified text blocks are actually moved to the place where the corresponding text is assumed to have been dictated. The moved regions are then realigned, such that the result of this (and the previous) step is a new alignment column.
- **Application of reconstruction rules**:
  Reconstruction rules specified by the user are applied to this alignment, and two additional columns are created: one containing the reconstructed words and another one consisting of a justification (i.e., the responsible rule) for that reconstruction.

written text	↔	*recognised text*
HEART:	=	Heart
Examination	=	examination
is	<	
normal	COR	normal
.	COR	.
LUNGS:	=	Lungs
	>	are
Clear	=	clear
.	COR	.

Fig. 7. Excerpt of aligned input sequences with sliding rule window indicated by solid frame

- **Reconstructing moved blocks**:
  Finally, the moved parts of the report are reinserted in order to resemble the original input.

6.1 Rule Engine

Once a stable alignment has been established, knowledge about corresponding passages can be used for inspecting tokens and their contexts both in the edited document and the output of the speech recognition system.

For this purpose, a rule engine has been developed. The reconstruction rules that are interpreted by this engine provide a mechanism for inspecting a sliding window that is moved over multiple columns according to their alignment. In addition to columns for the edited document (cf. figure 7, left side) and the output of the speech recogniser (cf. figure 7, right side), a so-called "alignment" column is available that indicates the correspondence between the left and the right side at the current element: "=" indicates that some kind of similarity has been found between the left and the right side, and therefore a substitution has been performed, whereas "<" indicates a deletion and ">" indicates an insertion. In the case of deletion, there is no element on the right side corresponding to that on the left side. Symmetrically, there is no element on the left side if the alignment column contains an insertion label.

Figure 7 depicts aligned columns and the sliding window of a rule that is used to inspect the column elements and their context at a certain position in the input. For each rule, a regular expression is applied to the alignment column, which specifies a dynamic sliding window size. In the example above, the regular expression might have been formulated in such a way that the sliding window iterates over instances of consecutive lines labelled with "=", with the intention of inspecting only whole blocks of elements for which some kind of similarity has been found.

Each rule adheres to the following skeleton:

```
rule rulename
match -w/window regexp/
 # inspect sliding window
do
 # specify reconstruction result
done
```

As explained above, the "window regular expression" works on the string of labels in the alignment column and specifies for which lines the rule window is set up. The `match` block can then be used to inspect all columns within the borders of the window. If the rule finds that the lines inside the window exhibit a phenomenon that this rule can handle, a non-zero value is returned in the `match` block, which causes the `do` block to be triggered. The `do` block is then responsible for building a literal transcription of the matching lines and writing it to a result column.

The advantage of this approach is that each phenomenon (like recogniser errors, repetitions, etc.) can be handled by a separate rule which encapsulates both the detection of such cases as well as the required knowledge to decide which column should be used or which transformations have to be applied to build an appropriate literal transcription for the current window.

The bodies of the `match` and `do` blocks can be freely expressed in regular Perl code. In addition, some special built-in functions for measuring phonetic and semantic similarity between two strings, and for converting formatted expressions into their most likely spoken variant (e.g.: `500 mg` → `five hundred milligrams`, cf. section 5.2.1) are available in these blocks.

Since more than one rule can match for a certain sequence of alignment labels, rules match on a first-come first-serve basis, meaning that rule precedence influences the result. In the experiments (cf. section 7), the effect of rule ordering is investigated explicitly.

written text	↔	*recognized text*	**CTX**$_{pho}$	**OVG**$_{pho}$	**OVS**$_{pho}$	*reference text*
a	COR	a	a	a	a	a
Char·cot	=<	sharp	Charcot	Charcot	Charcot	Charcot
	>=	cold				
foot	=	foot	foot	foot	-	foot
,	<		-	,	-	
though	=	no	though	-	-	though
there is	COR	there is	there is	there is	there is	there is
no	COR	no	no	no	no	no
ul·ce·ra·tion	====	al·te·ra·tion	ulceration	ulceration	-	ulceration
	<<<=	in			-	
of	<		-	of	-	of
skin	COR	skin	skin	skin	skin	skin

Fig. 8. A sample alignment containing two mismatch regions (dashed boxes), together with reconstruction rule results. Syllable boundaries are marked with dots [·]. Note that the [=]-overlap symbol just indicates correspondence, not equality of syllables, in contrast to the insertion [<] and deletion [>] symbols which label non-matching syllables. The solid boxes highlight lines affected by each rule, dashes [−] mark parts not covered by the rule.

6.2 Rule definitions

To test the effects of the previously described techniques, we specified reconstruction rules, where an alignment label is either the identity edit operation (COR) or a sequence of alignment labels [=, <, >] (cf. figure 8). The rules can be grouped into three categories: baseline rules, semantics-based rules, and phonetics-based rules.

Baseline rules are the three simple starting points for the hypothesized reconstruction that do not require any advanced processing:

- **Baseline**: only identical words in the alignment (COR) are reconstructed, mismatch regions are ignored.
- **Recognised-only (REC)**: for each alignment label, always select the *recognised text* for reconstruction.
- **Written-only (WRI)**: for each alignment label, always select the *written text* for reconstruction.

Semantics-based rules implement semantic knowledge in the reconstruction process. With regard to the initial assumption that reformulations are semantically similar, semantic rules select the *recognised text* for reconstruction, as soon as the rule matches.

Phonetics-based rules on the other hand try to detect corrections of speech recognition errors in the alignments. Therefore, they select the *written text* for reconstruction whenever the rule matches. As recognition errors are more likely to occur than reformulations, these rules should match more often than the semantic rules.

The following types of rules were defined for both semantic and phonetic similarity separately as indicated by subscripts in section 7:

- **Context (CTX)**: matches sequences of 1, 2, or 3 alignment labels containing at least one submatching label (=), if similarity is higher than threshold t. The idea behind this rule is that longer corresponding regions in the alignment are more likely to be real correspondences.
- **Overlap, greedy (OVG)**: matches sequences of 2 or 3 alignment labels, where inserted or deleted submatching labels (</>) are either preceded or succeeded by at least one matching label (=), if similarity is higher than threshold t. This rule collects all word sequences showing any possible overlap at submatching level without regard of the matching order.
- **Overlap, selective (OVS)**: matches sequences of 2, 3, or 4 alignment labels, where submatching labels (=) are first succeeded by insertion (<), and then preceded by deletion (>) labels if similarity is higher than threshold t. This pattern is typical for segmentation errors in the recognised text.

Figure 8 illustrates the effect of each rule on a sample alignment for the phonetic similarity case. The **context** rule is activated whenever a group of matching syllables appears. Still, it is not enough as it does not handle stand-alone insertions or deletions appropriately. The **greedy overlap** rule can handle insertions and deletions whenever they appear in terms of a syllable overlap. However, it is not activated when there is a direct match (`though` ↔ `no`). The **selective overlap** rule, finally, matches only the precise first segmentation error, where the syllable counts exactly match. Accidental matches are therefore impossible. This example indicates that combination of rules may be beneficial.

7 Experiments

The text reconstruction process was evaluated for two different tasks to examine the performance of the SPARC method. First, the quality of the reconstruction was tested. For this test, a literal transcription was reconstructed and compared to a manual reference transcription for a set of medical reports. We define the evaluation as a text retrieval task, because the results reflect how much of the original text can be reconstructed and how much of the reconstructed text is actually part of the original text. This test is a true performance measure of the system, without considering any particular application. The performance of the main components - semantic and phonetic similarity measurement, and text reconstruction - will be evaluated separately in section 7.1.

Second, the speech recognition performance using reconstructed texts is measured (cf. section 7.2). In this test, the language model of the speech recogniser producing the recognised texts was re-trained with the reconstructed texts and tested on an independent test set. This test is only an indirect performance measure and is intended to demonstrate the applicability and impact on the speech recognition process. For this reason, we decided to test with a commercially available ASR system instead of an academic one and did not perform specific parameter tuning to keep the results more independent from the actually used ASR system.

7.1 Reconstruction quality

For measuring reconstruction quality, we report results in terms of the metrics $Recall = \frac{|COR|}{|COR|+|MISS|}$, $Precision = \frac{|COR|}{|COR|+|WRONG|}$, and their harmonic mean $F1$, where |COR| is the number of reconstructed words with perfect correspondence in the reference text, |MISS| is the number of words in the reference text without correspondence in the reconstructed text, and |WRONG| is the number of reconstructed words without correspondence in the reference text (Van Rijsbergen (1979)).

The evaluation corpus consisted of 735 written and recognised texts of about 335,000 tokens, as well as manually transcribed reference texts for validation of the hypothesized reconstruction. The texts were selected such that they equally represent three ranges of average word error rates (WER) for the recognised text compared to a manual reference transcription. Hesitations and incomplete words were removed beforehand to avoid biased results.

7.1.1 Semantic and phonetic similarity measurement

The impact of semantic and phonetic similarity measurement is studied by evaluating semantic and phonetic reconstruction rules separately before they are joined in a single system. For the phonetic rules, previous results from Petrik and Pernkopf (2008a) are summarised here, while for the semantic rules and the joint system, entirely new results are presented.

We start with the evaluation of the semantic rules in table 2. The first group covers the baseline rules (Baseline, REC, WRI), while the CTX$_{sem}$, OVG$_{sem}$, and OVS$_{sem}$ systems of the second group represent semantic context and overlap. The combination of all rules is denoted by all$_{sem}$.

	5-13% WER			20-25% WER			40-45% WER		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Baseline	**100**	78.9	88.2	**99.9**	64.6	78.4	**99.5**	46.0	62.9
REC	83.3	**93.4**	88.1	79.0	85.7	82.2	66.7	71.9	69.2
WRI	92.8	93.1	**92.9**	89.9	**89.6**	**89.8**	85.9	**85.4**	**85.6**
CTX$_{sem}$	98.6	87.6	**92.8**	97.3	76.9	85.9	95.6	60.6	74.2
OVG$_{sem}$	99.7	80.2	88.9	99.4	66.6	79.8	98.7	47.9	64.5
OVS$_{sem}$	**99.8**	79.2	88.4	**99.7**	65.0	78.7	**99.0**	46.5	63.3
all$_{sem}$	98.6	**87.7**	**92.8**	97.2	**77.0**	**86.0**	95.5	**60.8**	**74.3**

Table 2
Reconstruction results in % for semantics-based rules (second block) in comparison to baseline systems (first block). Best results for each row grouping are boldface.

The recognised text (REC) is not a good starting point for reconstructing a literal transcription. Although the recall scores are comparable to the other methods, many errors stem from the recognition process, resulting in poor precision. The written text (WRI) is more reliable for the domain of medical dictations.

Using semantic context (CTX$_{sem}$) for reconstruction returns accurate results with higher precision than recognised-only (REC) or written-only (WRI) reconstruction and significantly higher recall than the baseline system. This holds even more for the overlap rules (OVG$_{sem}$, OVS$_{sem}$): whenever semantic overlap is detected, it is almost always correct. Unfortunately, the recall scores are only 0.4% - 2.0% absolute higher than the baseline scores, indicating a low number of matches for these rules. In sum, neither the separate semantic rule systems nor their combination is able to exceed the baseline systems for any of the WER ranges.

The threshold value for semantic similarity measurement can take values between $t = 0$ (no similarity) and $t = 7$ (identity) and was varied from $t = 1$ to $t = 7$ in the experiments. The resulting curves are plotted in a Recall/Precision diagram (cf. figure 9). Adjusting the semantic similarity threshold does not contribute much to the overall performance. The trade-off between recall and precision is almost linear, as is shown by the graphs in figure 9. The best recall/precision value pairs were obtained for a similarity threshold value $t = 5$ for all WER ranges.

Likewise, we evaluated the phonetic rules separately and in combination compared to the baseline systems. Table 3 summarises the results.

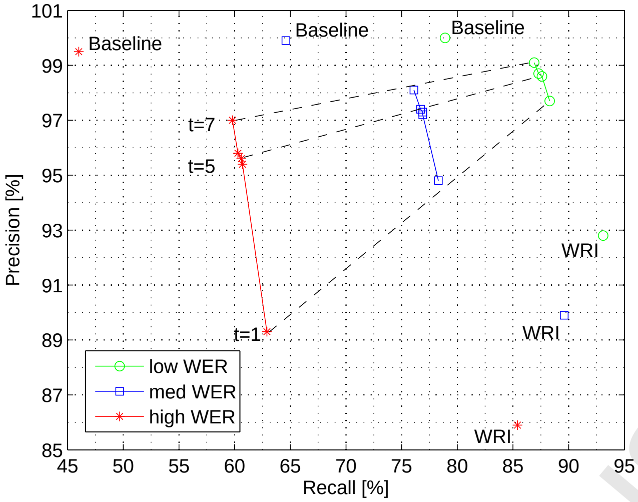


Fig. 9. Recall/Precision diagram derived from the all$_{sem}$ system by varying the semantic similarity threshold t between $t = 1$ and $t = 7$ for high, medium, and low WER texts.

	5-13% WER			20-25% WER			40-45% WER		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Baseline	**100**	78.9	88.2	**99.9**	64.6	78.4	**99.5**	46.0	62.9
REC	83.3	**93.4**	88.1	79.0	85.7	82.2	66.7	71.9	69.2
WRI	92.8	93.1	**92.9**	89.9	**89.6**	**89.8**	85.9	**85.4**	**85.6**
CTX$_{pho}$	97.6	90.4	93.8	95.4	82.8	88.7	93.1	69.8	79.8
OVG$_{pho}$	97.9	86.4	91.8	95.8	78.3	86.2	93.1	65.7	77.0
OVS$_{pho}$	**99.8**	79.5	88.5	**99.6**	65.6	79.1	**98.8**	47.3	64.0
all$_{pho}$	97.0	**91.1**	**94.0**	94.7	**84.3**	**89.2**	92.1	**72.6**	**81.2**

Table 3
Reconstruction results in % for phonetics-based rules (second block) in comparison to baseline systems (first block). Best results for each row grouping are boldface.

In the phonetically controlled reconstruction contextual information (CTX$_{pho}$) returned better F1 scores than in the semantically-controlled reconstruction. Only for the low WER case, however, a gain of 0.9% absolute can be observed in contrast to the written-only (WRI) reconstruction. The greedy exploration of overlap on syllable level (OVG$_{pho}$) returned surprisingly precise results which are absolutely comparable to using contextual information. This applies even more to the selective overlap rule (OVS$_{pho}$), which has only very little gain in recall in comparison to the baseline, but almost maximum precision. These findings indicate that the combination of these rules could be beneficial. The combination of all rules shows the best performance for all

WER ranges.

The threshold value for phonetic similarity measurement can be adjusted between $t = 0.0$ (no similarity) and 10.0 (identity) and was varied from $t = 5.0$ to 10.0 in the experiments. Like for semantic similarity measurement, the resulting curves are plotted in a Recall/Precision diagram, shown in figure 10. Optimising the threshold value for phonetic similarity also contributes to the overall performance. The trade-off between recall and precision is not linear, as the graphs in figure 10 show. The best recall/precision value pairs were obtained for a similarity threshold value $t = 8.0$ for all WER ranges.

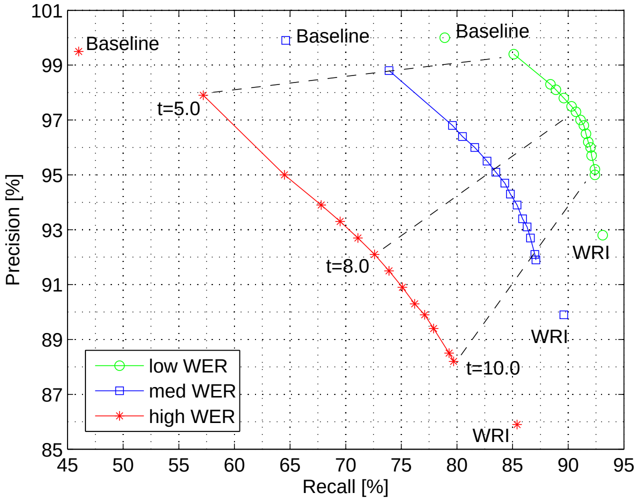


Fig. 10. Recall/Precision diagram derived from the all$_{pho}$ system by varying the phonetic similarity threshold t between $t = 5.0$ and $t = 10.0$ for high, medium, and low WER texts.

The SPARC method tries to combine knowledge about semantic and phonetic similarity to detect matches, corrections, and reformulations in the data (cf. section 2, figure 1). For this reason, the best semantics- and phonetics-based systems were combined into a single system. As mentioned before, the rule engine is sensitive to rule precedence, so there are several possible combinations. Thus, the impact of semantic and phonetic knowledge in the reconstruction process can be estimated. Table 4 lists the results for the given combinations: the I+S and I+P systems are combinations of the baseline and the all$_{sem}$/all$_{pho}$ systems, where the results are taken from tables 2 and 3, respectively. The I+S+P and I+P+S systems are combinations of the baseline, semantic and phonetic systems with the given rule precedence.

In terms of reconstruction performance, the combination of semantic and phonetic rules leads to improvements in recall without major losses in precision, resulting in gains in F1. The semantic system improves significantly (1.4% to 7.4% relative) while the phonetic system improves only slightly (0.1% to 0.25% relative). The best results are obtained when phonetic rules are given

	5-13% WER			20-25% WER			40-45% WER		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
I+P	97.0	91.1	94.0	94.6	84.2	89.1	92.1	72.5	81.1
I+S	**98.6**	87.6	92.8	**97.3**	76.9	85.9	**95.6**	60.6	74.2
I+P+S	97.0	**91.3**	**94.1**	94.5	**84.6**	**89.3**	91.9	**72.8**	**81.3**
I+S+P	97.9	90.5	**94.1**	95.9	82.7	88.8	93.4	69.5	79.7

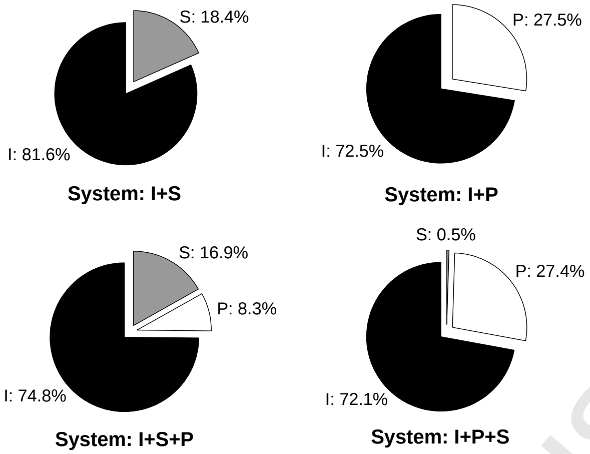
Table 4
Reconstruction results in % for combinations of the baseline identity (I), semantics- (S), and phonetics-based (P) rules. Rule precedence is indicated by the order of the rule addition terms. Best results for each column are boldface.

precedence over semantic rules. The detailed statistics on rule matching counts in table 5 explain this observation. In about 70 to 80% of all cases, identical items are detected which are matched by the baseline identity rule. Semantic rule matches account for about 18% of all matches and phonetic matches for about 28%, when applied separately to the alignments. In combination, however, phonetic rules still match in about 8% of all cases after semantic matching, while semantic rules only match in 0.5% after phonetic rules have been applied. Therefore, it can be concluded that 8% of mismatches are of pure phonetic nature, only 0.5% of pure semantic nature, and the rest of about 17 to 18% can be explained in both semantic and phonetic terms.

7.1.2 Rule-based vs. data-driven reconstruction

The rule-based reconstruction approach was compared to a data-driven approach to evaluate the classification performance. For data-driven text reconstruction, we use different classifiers to produce the hypothesized literal transcription which is the 2-class output of a classifier, i.e., either *written text* or *recognised text*. For classifier training, the class labels are produced by aligning the reference text with the written text. The features are derived from the automatic alignment and the phonetic similarity score (see section 3.1) computed for the aligned written and recognised phoneme strings. In addition, this score is derived for 3 consecutive phoneme strings to model the dependency of adjacent words in the classifier. The remaining features are computed from the sequence of submatching alignment labels. Therefore, the sequence is split into 3 equal parts. After assigning values to the labels, i.e. $[=]\ldots 0, [<]\ldots -1, [>]\ldots 1$, the mean and standard deviation of each part serve as feature. The last feature denotes the length of the syllable symbol sequence. Hence, 9 features are used for the classifiers. The following classification approaches are used (Bishop (2006)):

- ***k*-NN**: *k*-nearest neighbour classifier. For the presented results $k = 9$.



	TOTAL	identical (I)		semantic (S)		phonetic (P)	
	[1]	[1]	[%]	[1]	[%]	[1]	[%]
I+S	79,525	64,911	81.6	14,614	18.4	0	0
I+P	89,592	64,923	72.5	0	0	24,669	27.5
I+S+P	86,741	64,916	74.8	14,621	16.9	7,204	8.3
I+P+S	90,033	64,923	72.1	441	0.5	24,669	27.4

Table 5
Rule matching counts and percentages for the combined rule systems.

- **NN**: Neural network (Multilayer Perceptron) with 3 layers. The number of neurons in the input and output layer is set to the number of features and the number of classes, respectively. The number of neurons in the hidden layer is set to 70. We use Levenberg-Marquardt backpropagation for training, a hyperbolic tangent sigmoid transfer function for the neurons in the input and hidden layer, and a linear transfer function in the output layer.
- **SVM**: The support vector machine with the radial basis function (RBF) kernel uses two parameters C^* and σ, where C^* is the penalty parameter for the errors of the non-separable case and σ is the parameter for the RBF kernel. We set the values for these parameters to $C^* = 1$ and $\sigma = 1.5$.

The optimal choice of the parameters, kernel function, number of neighbours, and transfer functions of the above mentioned classifiers has been established during extensive experiments. Five-fold cross-validation is used to produce the results with the classifiers. Throughout our experiments, we use exactly the same data partitioning for each training procedure.

Table 6 lists the data-driven systems k-NN, NN, and SVM in comparison to the best combined rule-based system I+P+S. Both, rule-based and data-driven reconstruction use the same input features derived from the alignment

	5-13% WER			20-25% WER			40-45% WER		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Baseline	**100**	78.9	88.2	**99.9**	64.6	78.4	**99.5**	46.0	62.9
REC	83.3	**93.4**	88.1	79.0	85.7	82.2	66.7	71.9	69.2
WRI	92.8	93.1	92.9	89.9	**89.6**	89.8	85.9	**85.4**	**85.6**
I+P+S	97.0	91.3	**94.1**	94.5	84.6	89.3	91.9	72.8	81.3
k-NN	94.9	92.8	93.9	91.6	88.0	89.8	87.1	83.4	85.2
NN	94.9	93.0	93.9	91.4	88.5	**89.9**	86.6	84.0	85.3
SVM	94.8	93.0	93.9	91.3	88.6	**89.9**	86.6	84.4	85.5

Table 6
Reconstruction results in % for baseline systems (first block), the best rule-based system (second block), and data-driven systems (third block). Best results for each column are boldface.

labels, semantic, and phonetic similarity scores.

The data-driven systems are closer to the written text only (WRI) reconstruction than the rule-based system, showing improvement in precision for all WER ranges. The rather simple *k*-NN classifier consistently produces the highest precision while the more complex NN and SVM classifiers achieve higher recall scores. The rule-based system outperforms the data-driven system only for low error rates.

The selection of either rule-based or data-driven reconstruction framework depends on the intended application. The definition of rules allows the precise control of the reconstruction process and specific fine tuning for either high precision or high recall. Furthermore, it can be used efficiently for "labelling" a corpus of parallel recognised and written texts by applying specific rule configurations. The data-driven system, however, is better when the amount of reconstructed data needs to be maximised, particularly for the high WER condition. The main benefit is then that no handcrafting of rules and no tuning of similarity thresholds is required.

7.2 Automatic Speech Recognition

We demonstrate the application of reconstructed texts in an attempt to improve the automatic speech recognition system with which the input recognised texts were produced. For this exemplary evaluation, we retrained the language model of this commercial backend speech-recognition system for telephone channel audio with reconstructed texts. The SPARC method was compared to the standard method for language model training and to a random generation of reconstructed text.

The reconstruced texts could as well be utilised for retraining the acoustic models, e.g., by providing improved transcripts of audio training data. We expect the gains for this task, however, to be low, since the amount of material that the SPARC method makes available for acoustic training in addition to the material that is already there is relatively small.

The evaluation presented in the following not only measures the performance of the modified ASR system, but also shows in an exemplary fashion how for any other application an appropriate SPARC reconstruction approach is selected based on the findings from section 7.1.

7.2.1 Language modelling approach

The standard approach of this ASR system for creating language models is to segment large corpora of written text into lexicon entries and to train trigram models on them. The mapping of written text onto lexicon entries is not a trivial process since formatted items like numbers, quantities, dates etc. cannot be directly represented as lexicon entries and give only little clue as to what a speaker would say to dictate such items. Written text further contains additions (including punctuation marks) and reformulations by transcriptionists and, therefore, does not represent what actually has been or will be dictated.

To handle the common cases of formatted numeric items ("grammar items") and inserted punctuation marks, a 2-stage decoding-rescoring strategy is applied:

- **Decoding:** An initial language model is trained on regular words and classes of grammar items. This language model cannot cover spoken forms of grammar items. Therefore, at decoding time, it is interpolated with an additional language model derived from grammars representing spoken forms of grammar items ("grammar language model").
- **Rescoring:** The emerging wordgraph is parsed for grammar items, and enriched with edges marked with the corresponding grammar classes. This wordgraph then is rescored using the initial language model. At the same time, punctuation marks are hypothesized.

The setup is robust in language-model adaptation as only the class of a grammar item needs to be determined, without guessing a word sequence that might have been spoken. Its weakness, however, lies in the imprecise grammar language model used at decoding time.

7.2.2 Experimental setup

We chose the domain of Clinical Reports as test case since there was enough data available, both to create reconstructed text (requiring written and recognised text) and to evaluate the performance (requiring reference transcriptions of what has actually been dictated). Our setup reconstructed 101,607 reports from 504 authors (between 45 and 507 reports per author) running in parallel on four standard personal computers for three weeks and produced a text corpus of 52 million words (cf. table 7), the equivalent of about 9,000 hours of sound.

The SPARC reconstruction approach was selected as follows: The baseline word error rate of the chosen corpus indicated a low-to-medium WER condition. According to table 6 either the rule-based system with I+P+S rules or one of the data-driven reconstruction approaches appeared to be promising for text reconstruction. Previous language model adaptation experiments conducted on less data showed that a certain fragmentation is introduced in the reconstructed text by the data-driven reconstruction approaches resulting in higher language model perplexities (Petrik and Pernkopf (2008b)). Therefore, we selected the rule-based I+P+S system for text reconstruction in the experiments.

The speaker-independent acoustic models of the ASR system have been trained on 200 hours (female speakers) and 300 hours (male speakers) of acoustic material recorded on a telephone channel with 4kHz bandwidth, Acoustic speaker-adaptation was performed using Maximum-Likelihood Linear Regression (MLLR) for the first 15 minutes of sound, and Maximum-A-Posteriori (MAP) adaptation for the rest of the available data, which was 10 hours of sound for each speaker in the test set.

For this domain, a large medical lexicon of 58,103 words was used, giving a high coverage on both the test set and the training corpus (OOV rates < 1.5%).

Recognition tests were performed on a set of 239 reports from 2 female and 3 male authors (3 hours of sound per user), all in the domain of Clinical Reports[4], and all recorded through a telephone channel with 4kHz bandwidth. The best available acoustic references for these speakers were used. The baseline word error rate on our test set was 11.77% (cf. table 8).

[4] The available speakers for the domain of Clinical Reports do not cover all word error rate ranges of the previous experiments in section 7.1. For this reason, results are reported per speaker and on average, but not according to the previous separation into low, medium, and high word error rate conditions.

7.2.3 Language models

The reconstruction of spoken words allows to avoid the use of language model classes for grammar items. To measure this effect and the quality of the resulting language models, we perform recognition tests with four different trigram language models:

(1) **Broad:** A language model used in commercial applications, created from 287 million words of general medical reports. This language model requires an interpolated grammar language model at decoding time, since it trains grammar items as classes.
(2) **Classic:** This language model is built from the corrected text of the reconstruction corpus (52 millions words), and is otherwise consistent with initial language models used in commercial applications, requiring an interpolated grammar language model at decoding time.
(3) **Random as-spoken:** Same as classic, but instead of grammar classes, a randomly chosen spoken representation of that class is trained. This is a standard technique to get closer to what has been spoken, at least in the case of grammar items, and can be seen as a simple case of reconstruction. It does not require a grammar language model at decoding time any more.
(4) **Reconstructed:** Reconstructed text produced by SPARC was slightly post-processed to match the lexicon: Phrases (i.e., multi-word expressions handled as single lexicon entry like "she is", "he had" etc.) are handled by SPARC as word sequences and were mapped back to single lexicon entries. Special words for punctuation marks were reintroduced (SPARC reconstructs dictated periods and commas as lexicon entries "period" and "comma", while the lexicon and the rescoring language model use special symbols).

Language model	size	OOV rate
Broad	287M	0.9%
Classic (baseline)	51.92M	1.15%
Random as-spoken	56.37M	1.15%
Reconstructed	49.64M	1.34%

Table 7
Language model details: Number of tokens and out-of-vocabulary (OOV) rate.

The chosen evaluation method is biased against the SPARC reconstruction approach and leads to slightly worse results for two reasons:

- SPARC reconstructs the so called "demographic header", this is demographic patient information at the beginning of the dictation, which is not a part of the final report. Recognition performance on the demographic header is ruled out in all tests since recognition accuracy in this section is of no benefit for the user.

- The lexicon (and rescoring language model) makes a distinction between special words like "Lungs:" and "Heart:" versus "lungs" and "heart", respectively to be able to produce appropriately formatted output. SPARC always reconstructs the regular words; therefore, recognition accuracy is expected to be lower on these special words.

The information recovered by the SPARC method may also be included in other ways into language model training. Instead of reconstructed text, an annotated database of particular observations such as spoken forms of grammar items or other reformulations may be generated. An extensive evaluation of methods for including this annotated data in language model training as, e.g., embedded grammars or language model classes was, however, not the focus of this work.

7.2.4 Results

The results are summarised in table 8. Using the reconstructed text language model reduced the overall word error rate from 11.77% to 10.86% which is a relative reduction of 7.74% compared to the baseline classic language model. The randomly generated as-spoken variants only lead to an overall relative reduction of 4.38%. Table 8 also shows that these improvements are consistent for all speakers. Both, the random as-spoken and reconstructed text language models even outperform the broad language model which was created from substantially more data. Hypothesizing the spoken forms of grammar items is therefore beneficial for the applied 2-stage decoding-rescoring strategy.

Speaker	Broad	Classic	Random		Reconstructed	
	WER	WER	WER	rel. △	WER	rel. △
F1	7.68	8.25	7.93	-3.91	7.81	-5.36
F2	16.80	17.64	16.37	-7.18	15.42	-12.56
M1	16.79	17.37	16.76	-3.50	16.56	-4.67
M2	6.86	7.13	7.12	-0.28	6.57	-7.98
M3	9.33	8.87	8.46	-4.60	8.45	-4.80
Total	11.34	11.77	11.25	-4.38	10.86	-7.74

Table 8
ASR results for the tested language models: Word error rate (WER) and relative difference (rel △) to the Classic (baseline) model in [%].

Based on the findings from this first experiment, we conducted a second experiment where we gradually increased the corpus size for language model training from 1 million tokens up to the maximum size of about 50 million tokens. Figure 11 illustrates the evolution of the word-error rate with respect to the size of the language model training corpus. Up to a corpus size of 3 million words, there is not much difference between the models built on randomly generated as-spoken variants and reconstructed text. For a corpus size of 6 million tokens or more, the SPARC method performs consistently better. At 50 million words, the word-error rate begins to go into saturation, so incorporating more data will only have minor effects on the word-error rate.

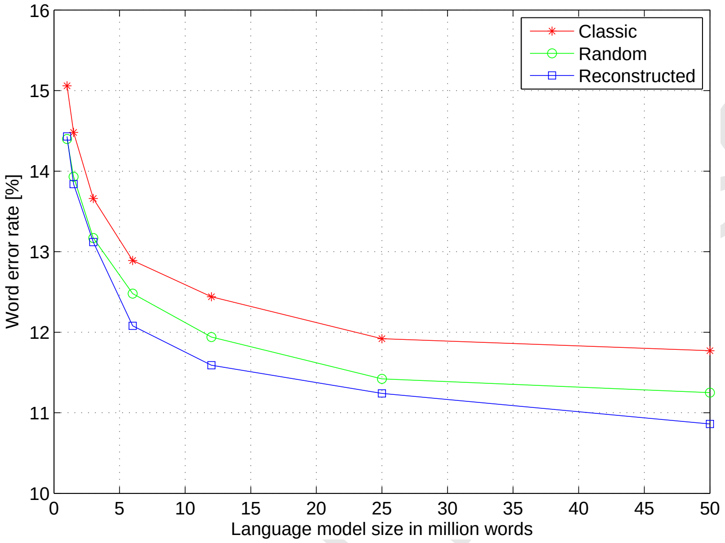


Fig. 11. WER in [%] for increasing re-training text corpus size.

Apart from the mentioned adjustments, the reconstructed texts were not further optimised or tuned for ASR purposes. Using the SPARC method without any further tuning immediately resulted in the reported improvements. Additional fine-tuning in terms of, e.g., the interpretation of punctuation or the exclusion of leading and trailing irrelevant text blocks in recognised texts may even further improve the performance.

8 Conclusion

We have described the SPARC method of semantics and phonetics based similarity measurement for the automatic reconstruction of medical dictations from draft recognised texts and final written reports. The resulting reconstructed texts can be used for various applications in language technology, including but not limited to acoustic and language model adaptation for automatic speech recognition, computer-aided document production in medical transcription, or generally for the development of parallel text corpora of non-literal text resources.

The method is based on an alignment between a draft speech recognition transcript containing errors and a formatted, corrected medical report that may have been paraphrased during the transcription process. The text alignment uses a model of semantic and phonetic similarity to detect corresponding (matching) regions in the texts and to properly align them on multiple levels of segmentation. For this purpose, semantic and phonetic similarity measures were developed for the matching procedure. The resulting alignment is interpreted with a newly developed rule engine which allows precise control over the reconstruction process with context-sensitive reconstruction rules.

The experimental evaluation showed that the text quality improved for the reconstructed text in comparison to both recognised and written text. For recognised texts with a low word error rate, the best reconstruction system improved the F1-score of the best baseline system from 92.9 to 94.1%. In general, phonetics-based rules proved to be more effective than semantic-based rules while semantics-based rules turned out to be more precise. Combining phonetic and semantic knowledge for text reconstruction improved the reconstruction quality. A more detailed analysis revealed that 8% of the resolved mismatches are of pure phonetic nature, only 0.5% of pure semantic nature, and about 17-18% are detectable with both semantic and phonetic measures together. The rule engine for reconstruction proved to have comparable performance to a data-driven classification system for the low word error rate condition, while for medium and high word error rates, the automatic classifiers returned better results.

Concerning the overall benefit for the speech recognition system, an experiment with a retrained language model based on reconstructed texts yielded a word error rate reduction of 7.74% relative in comparison to a standard retraining, and of 4.38% relative for a language model based on randomly reconstructed text. As no specific optimisation has been performed yet, further improvements by parameter tuning are still possible.

In future work, we plan to complement our experiments by evaluating the usage of reconstructed text for retraining of the acoustic models. Furthermore, we want to compare our results to different approaches of including semantic information into the language model, e.g., as classes or embedded grammars.

The focus of the SPARC approach on the assignment of phonetic and semantic similarity between aligned speech recognition results and final reports has turned out to be useful and suitable for the reconstruction of literal transcripts. Three main aspects of the approach have already turned out to be beneficial: 1) Reconstructed texts reduce the required amount of manually transcribed texts for training of speech recognition systems. 2) Retraining with reconstructions leads to slightly lower word-error rates in speech recognition. 3) Since the reconstruction and alignment are knowledge based, our methods are already being used as annotation tools for semantic and phonetic information and serve as a starting point for automatic document creation.

9 Acknowledgements

This research has been carried out in the context of the projects SPARC and COAST (http://www.coast.at). We gratefully acknowledge funding by the Austrian FIT-IT and K-net programs, ZIT Zentrum fuer Innovation und Technologie, Vienna, Steirische Wirtschaftsfoerderungsgesellschaft mbH and Land Steiermark.

The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry for Transport, Innovation, and Technology and by the Austrian Federal Ministry for Science and Research.

References

Bishop C., 2006. *Pattern Recognition and Machine Learning*, Springer.

Burgun, A. and Bodenreider, O., 2001. Comparing terms, concepts and semantic classes in WordNet and the Unified Medical Language System. In *Proceedings of NAACL'2001 Workshop, "WordNet and Other Lexical Resources: Applications, Extensions and Customizations"*, pp. 77–82.

Chen, S.F. and Goodman, J., 1996. An Empirical Study of Smoothing Techniques for Language Modeling, In *Proceedings of the ACL*. Santa Cruz, California, pp. 310–318.

Fellbaum C. (ed.), 1998. *WordNet: An Electronic Lexical Database*, MIT Press.

Filali, K., Bilmes, J., 2005. A dynamic Bayesian framework to model context and memory in edit distance learning: An application to pronunciation classification. In: *Proceedings of the ACL*. Ann Arbor, Michigan, pp. 338–345.

Fosler-Lussier, E., Amdal, I., Kuo, H.-K. J., 2005. A framework for predicting speech recognition errors. *Speech Communication* vol. 46, pp. 153–170.

Hammond, M., 1995. Syllable parsing in English and French. Rutgers Optimality Archive, http://roa.rutgers.edu/.

Hazen, T. J., 2006. Automatic alignment and error correction of human generated transcripts for long speech recordings. In *Proceedings of the ICSLP*. Pittsburgh, Pennsylvania, pp. 1606–1609.

Huber M., Jancsary J., Klein A., Matiasek J., Trost H., 2006. Mismatch interpretation by semantics-driven alignment. In *Proceedings KONVENS-2006*, Universität Konstanz.

Jancsary J., Klein A., Matiasek J., Trost H., 2007. Semantics-based Automatic Literal Reconstruction Of Dictations. In In *Workshop on the Semantic Rep-*

resentation of Spoken Language (SRSL07), CAEPIA - TTIA, Salamanca, Spain.

Kemp, T. and Waibel, A., 1999. Unsupervised training of a speech recognizer: recent experiments. In *Proceedings of EUROSPEECH'99*, Budapest, pp. 2725–2728.

Lamel, L., Gauvain, J.-L., Adda, G., 1998. Lightly supervised and unsupervised acoustic model training. In *Computer Speech & Language*, vol. 16, pp. 115–129.

Levenshtein, V., 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics - Doklady* vol. 10, pp. 707–710.

Lindberg D.A.B, Humphreys B.L., McCray A.T., 1993. The Unified Medical Language System. *Methods of Information in Medicine*, vol. 32, pp. 281–291.

Moore R. K., 2003. A Comparison of the Data Requirements of Automatic Speech Recognition Systems and Human Listeners. In *Proceedings of EUROSPEECH'03*, Geneva, pp. 2582–2584.

Pakhomov, S., Schonwetter, M., Bachenko, J., 2001. Generating training data for medical dictations. In *Proceedings of the NAACL*, Pittsburgh, Pennsylvania.

Pedersen T., Pakhomov S.V.S., Patwardhan S., and Chute Ch.G., 2007. Measures of Semantic Similarity and Relatedness in the Biomedical Domain. *Journal of Biomedical Informatics*, vol. 40, pp. 288–299.

Peters, J., Drexel, C., 2004. Transformation-Based Error Correction for Speech-to-Text Systems, *Proceedings of the ICSLP*, Jeju Island, South Korea, pp. 1449–1452

Petrik, S. and Kubin, G., 2007. Reconstructing medical dictations from automatically recognized and non-literal transcripts with phonetic similarity matching. In *Proceedings of the ICASSP'07*, Honolulu, Hawaii, pp. 1125–1128.

Petrik, S. and Pernkopf, F., 2008a. Automatic Phonetics-Driven Reconstruction of Medical Dictations on Multiple Levels of Segmentation. In *Proceedings of the ICASSP'08*, Las Vegas, Nevada, pp. 4317–4320.

Petrik, S., Pernkopf, F., 2008b. Language model adaptation for medical dictations by automatic phonetics-driven transcript reconstruction. *Proceedings of the IASTED Intl. Conf. on Artificial Intelligence*, Innsbruck, Austria, pp. 194–199.

Prince, A. and Smolensky, P., 2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell.

Printz, H., Olsen, P. A., 2002. Theory and practice of acoustic confusability. Computer Speech and Language 16, 131–164.

Rastrow, A., Sethy, A., Ramabhadran, B., 2009. A new method for OOV detection using hybrid word/fragment system. In: Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP). Taipeh, Taiwan, pp. 3953–3956.

Ristad, E. S., Yianilos, P. N., 1998. Learning String-Edit Distance. In *IEEE*

Trans. Pattern Analysis & Machine Intelligence, vol. 20, pp. 522–532.

Stolcke, A., 2002. SRILM - An Extensible Language Modelling Toolkit. In *Proceedings of the ICSLP*, Denver, Colorado, pp. 901–904.

Van Rijsbergen, C.J., 1979. *Information Retrieval*, Butterworths.

Voll, K.D., 2006: *A Methodology of Error Detection - Improving Speech Recognition in Radiology.* PhD Thesis, School of Computing Science, Simon Fraser University, Canada.

Zobel, J., Dart, P. W., 1996. Phonetic string matching: Lessons from information retrieval. In *Proceedings of the 19th International Conference on Research and Development in Information Retrieval*, Zurich, Switzerland, pp. 166–172.