

Semantics-based Automatic Literal Reconstruction Of Dictations

Jeremy Jancsary¹, Alexandra Klein¹, Johannes Matiasek¹, and Harald Trost²

¹ Austrian Research Institute for Artificial Intelligence
A-1010 Wien, Freyung 6/6
firstname.lastname@ofai.at

² Institute of Medical Cybernetics and Artificial Intelligence
of the Center for Brain Research, Medical University Vienna, Austria
Harald.Trost@meduniwien.ac.at

Abstract. This paper describes a method for the automatic literal reconstruction of dictations in the domain of medical reports. The raw output of an automatic speech recognition system and the final report edited by a professional medical transcriptionist serve as input to the reconstruction algorithm. Reconstruction is based on automatic alignment between the speech recognition result and the edited report. Based on an ontology (i.e. UMLS) and lexical resources (i.e. WordNet and an inventory of spoken variants for each concept), semantic representations are assigned to terms and phrases. Alignment takes into account semantic similarity scores, based on the similarity between semantic representations of the two sources, and phonetic similarity scores. This paper explains how the speech recognition output is compared and aligned to the edited written documents and how the two different input sources are complementary for the task of reconstructing a literal transcript.

1 Introduction

Literal transcription corpora are a valuable resource in the domain of automatic speech recognition. First, they can serve as input for initial training or improvement of the language model of a speech recognition system. Second, future dictation systems aim at moving away from creating written drafts to producing structured documents that conform to the formal and informal requirements of their respective domains. Currently, the output of dictation systems is edited by trained typists in order to obtain a satisfactory document. Depending on the experience of the speaker, utterances must be expanded, restructured or reformulated to conform to the required conventions for written reports. Automating these tasks is the focus of ongoing research. Again, high-quality literal transcripts that are fairly free from recognition errors are a prerequisite for learning transition rules between spoken language and written reports.

However, manually creating literal transcripts is a very laborious and hence cost-intensive task. It is therefore desirable to use available resources to automatically create transcripts that closely match the quality of manually created literal ones.

The method presented in this paper narrows the quality gap between automatically and manually created transcripts. For this purpose, it uses two resources that are readily available in many scenarios:

- Transcripts created by an automatic speech recognition system. These are ridden with the usual errors introduced during speech recognition (confusion of homophones, etc.).
- Edited reports that are created out of above-mentioned transcripts by professional typists. The resultant documents are usually free of errors introduced by the speech recognition system; however, they still deviate from a literal transcription in that the typists perform some paraphrasing or even structural changes on the raw document.

The task is then to exploit the complementary potential of these two resources: In theory, recognition errors in the raw output of the speech recognition system can be found by comparing the textual units to those in the edited document. Similarly, rephrasings in the edited document can be detected because the raw output of the speech recognition system stays close to the dictation (modulo recognition errors).

When inferring a literal transcription from the input (“reconstruction”), the following basic strategy for handling corresponding, yet not completely equal, passages of the two input documents will be applied:

Passages where the edited document and the raw output of the speech recognition system are phonetically similar will be considered recognition errors. Therefore, the literal transcription for that passage will be built from the edited document. If, on the other hand, corresponding passages in the edited document and in the raw output of the speech recognition system are semantically similar, we can assume that the typist performed some paraphrasing on the raw document. In this case, the raw document is a better candidate for producing a literal transcription.

2 Alignment

Establishing proper alignment of the edited document and the raw output of the speech recognition system is an important prerequisite for all further steps. During alignment, both input documents are viewed as sequences of tokens, where – in our approach – tokens are determined by concept rather than by word boundaries.

As an example, consider a numerical expression like `ninety-six over seventy`. The same expression might occur as `96/70` in the edited document. For comparison, it is helpful if both character sequences are grouped into a single token each. It is considerably harder to identify that `acute myocardial infarction` and `AMI` refer to the same concept if the first expression is split over multiple tokens. Such tokenization is achieved via multiple finite state transducers that are compiled from both lexical knowledge and domain-specific grammars that handle numerical expressions, dates, units, etc. Tokenization via finite state transducers is described in [3] in detail.

A generalized Levenshtein algorithm [4] is then applied to the token sequences of the edited document and the speech recognition result. The alignment problem can be defined as a minimization problem, where a number of actions (i.e. substitutions, deletions and insertions) with associated costs can be performed to navigate through the search space. The costs are specified in a scoring function.

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

here that two passages denote the same section in the actual dictation. Naturally, the two passages need not necessarily consist of the same tokens. 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. More global comparisons are performed at the reconstruction stage (see section 3).

2.1 Modelling semantic similarity

In order to measure semantic similarity in a semantic scoring function, resources that map words onto some kind of semantic representation are needed. Since our application domain is medical reports, medical terminology has to be incorporated into the system. The resource we employ for that purpose is the Unified Medical Language System (UMLS, [6]), that comes with a metathesaurus, a semantic network and a lexicon (SPECIALIST) – for an overview of semantic similarity assignment in the medical domain, cf. [5]. The morphosyntactic information from the lexicon was worked into the finite-state transducer that is used as a morphological lexicon and for tokenization. The metathesaurus is a very large, multi-purpose, and multi-lingual vocabulary database that contains information about biomedical and health related concepts, their various names, and the relationships among them. All concepts in the metathesaurus are assigned to at least one semantic type from the UMLS semantic network.

The WordNet lexical database [2] was used as domain-independent resource. For our purpose, the WordNet hypernym relation is the most important synset relation. The relations connecting WordNet synsets are quite different from the relations between UMLS concepts. For a study comparing WordNet and UMLS in greater detail, cf. [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)	2 same UMLS semantic type
6 same root (only inflection)	or <code>parent(word1, word2)</code>
5 synonymous	or <code>parent(word2, word1)</code>
4 derived	1 direct hierarchical relation between semantic types
3 siblings	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 check for a measure that is available in both WordNet and UMLS, has a finer granularity than the (rather crude) UMLS semantic type and 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.

The semantic score is compared to the phonetic score. For phonetic scoring, a mechanism was used which was developed at TU Graz for evaluating the phonetic distance of two token sequences [7]. This scoring mechanism measures phonetic distance: The closer the two argument token sequences are from a phonetic point of view, the lower the distance value that is returned.

3 Rule Engine

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

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 (the left side) and the output of the speech recognition (the 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. Figure 1 depicts aligned columns and the sliding window of a rule that is

edited document	alignment	recognizer output
...
Mr.	=	The
Stone	=	Bone
	>	is
	>	um
is	=	is
a	=	a
...

Fig. 1. Aligned input sequences with sliding rule window marked by horizontal lines

used to inspect the column elements and their context at a certain position in the input. For each rule, a regular expression working on the alignment column may be specified that controls which lines fall into the sliding window. 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 labeled with “=”, with the intention of inspecting only whole blocks of elements for which some kind of similarity has been found.

In addition to the preliminary regular expression check, a rule body can be freely expressed in regular Perl code and may be used to inspect not only the alignment column, but also the columns of the edited document and the speech recognition output. The rule body is also responsible for building a literal transcription of the matching lines. 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) are available for this purpose.

The advantage of this approach is that each phenomenon (like recognizer 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.

A rule base was defined by specifying rules manually and by an automatic mechanism which deduces rules from frequently occurring deviations.

4 Experiments

In this section, some selected phenomena will be discussed in detail, and the effectiveness of the reconstruction mechanism described in the previous sections will be evaluated for these phenomena. The parallel corpus of raw speech recognition output and corresponding edited documents which is used for these experiments was gained from report dictations of various medical fields that were first fed into a speech recognition system and then edited by trained medical typists. In addition, for a subset of these documents, a manual literal transcript was available which could be used for evaluating the accuracy of reconstruction.

4.1 Synonyms and Acronyms

Replacing terms or phrases by synonyms or acronyms is one of the main causes of deviations between the edited documents and the actual dictations. In our application – medical reports – words or phrases may be changed by the transcriptionists, mostly for stylistic reasons.

Detecting synonyms and acronyms is implemented mainly by using the resources described in subsection 2.1. If a synonym or an acronym is recognized, the respective reconstruction rule will choose to build the literal transcript for this token using the output of the speech recognition system (because it is assumed that the final document was edited for stylistic reasons).

In order to assess the quality of the reconstruction for synonyms and acronyms, we manually evaluated the reconstruction for a test set.

absolute token counts		performance metrics	
correct reconstruction	36	precision	0.900
wrong reconstruction	4	recall	0.450
not found	44	F1	0.600

The table above lists three important counts: First, it shows how many tokens were identified as repetitions, and thus lead to proper reconstruction of the literal transcript (*correct reconstruction*). Second, it shows how many tokens that were identified as repetitions lead to wrong reconstruction of the literal transcript (*wrong reconstruction*). Finally, it shows how many tokens should have been, but were not identified as repetitions and thus not successfully reconstructed (*not found*).

These numbers were determined manually on a set of 24 report dictations so that false negatives could also be found (*not found*). The three counts then allow for calculation of precision, recall and the harmonic mean (F1). The respective values show that the rule leads to fairly reliable reconstruction (about 90% of reconstructed tokens are correct).

We can conclude from these numbers that recognizing synonyms and acronyms leads to quite precise reconstruction. Coverage, however, could still be improved. Part of the problem is that expanded acronyms and some synonyms extend over multiple words. If tokenization fails to return them as one single token, it is considerably harder to map a term to the corresponding term in the other input sequence.

4.2 Repetitions

One inherent phenomenon of spoken language is that speakers tend to repeat themselves. Often, such repetitions are intermingled with hesitations and other phenomena. Figure 1 shows one typical example where a speaker first hesitates (“um”), and then repeats herself or himself (“is”). Frequently, longer regions are repeated, sometimes slightly deviant from the first utterance.

If the alignment of the edited document and the speech recognition output contains a passage that is labeled as inserted (“>”), the reconstruction rule checks the immediate context of that passage for a similar region that matches a passage in the edited document. This heuristic (enhanced with handling of some special cases) turns out to work fairly well. The numbers below can be interpreted in the same way as those for synonyms and acronyms (see subsection 4.1):

absolute token counts		performance metrics	
correct reconstruction	135	precision	0.794
wrong reconstruction	35	recall	0.498
not found	136	F1	0.612

About 80% of the reconstructed tokens are correct, but more than half of the repetitions are missing, mostly due to recognizer errors in repeated regions or their context.

4.3 Block Moves

Frequently, text appears at a different position in the written report than at the spot where it has been dictated. The reasons for that phenomenon are either a speaker error (e.g.: “please add to the medical history above ...”), or the transcriber herself decides to reorder dictated passages, e.g. for stylistic reasons or document structuring conventions. Such moved blocks present a considerable challenge for reconstruction, since the corrected report text corresponding to the dictation does not parallel the recognition result in the multialigned file.

In order to find moved blocks, we proceed as follows: First, the aligned report is scanned for sufficiently long, coherent regions that only have written columns and no recognizer columns. Then all regions are searched that only have recognizer columns. Finally, we compute the overlap between the flattened strings of each pair of regions.

Corresponding region pairs are considered as block moves, and the corrected report part is realigned with the recognizer results for the purpose of reconstruction. Overlap matching attempts to align the two words sequences and returns the optimal alignment in the form of an overlap string containing only the characters “<”, “=” or “>” (indicating deletion, insertion and match, respectively).

We took a set of 4800 Levenshtein prealigned reports for the evaluation of the block move detection algorithm and the assessment of its potential to increase the proportion of reconstructable items (remember that the reconstruction algorithm relies on optimally aligned sequences of written report text and recognizer results, which are not available in the case of moved text blocks). The parameter setup for block move detection was such that only blocks with a minimum length of 4 items, a minimum match count of 3 and a minimum match ratio of 25% were considered. We did no evaluation

absolute token counts			performance metrics		
	baseline	all rules		baseline	all rules
correct reconstruction	483356	539179	precision	0.998	0.959
wrong reconstruction	812	23002	recall	0.726	0.810
not found	182802	126280	F1	0.840	0.878

Table 1. Evaluation over 1637 reports

for recall, because these settings are per se rather relaxed and shorter or less corresponding text block pairs are of doubtful use for reconstruction anyway. What we did, however, was to evaluate for precision and the broadening of the reconstruction basis.

From the 4800 reports, 538 (11.21%) contained at least one block move, the distribution is shown in the table below:

	Number of Block moves				
	0	1	2	3	4
Reports	4262	487	46	4	1
Reports (%)	88.79	10.15	0.96	0.08	0.02

For 451 detected block moves, manual evaluation has been performed. 398 regions (88.25%) were fully correct, 44 regions (9.76%) were correct in their core content, but either the written or the spoken side contained some extra material at the region borders, so 442 extracted block moves (98%) were correct or almost correct, only 9 regions (2%) have been erroneously correlated. Repair phrases have been detected in 45 regions (10%). While the overall effect with respect to the total recognized items of the reports is rather small, the effect within reports containing moved blocks is more remarkable. In these cases, block move detection proves to be rather reliable and improves the reconstruction basis for reports containing such block moves by 4% on the average.

5 Findings and Summary

The previous section exemplarily discussed some phenomena, how these are dealt with, and how well the respective reconstruction rules work. For creating literal transcripts of complete reports, we reconstruct regions of a report for which literal transcripts can be produced at a good confidence level, and iteratively add high-precision rules to close the remaining gaps.

The quality of reconstruction can automatically be assessed by aligning the output of the reconstruction mechanism to manual literal transcripts of the reconstructed reports. It is then counted how many tokens were correctly reconstructed, how many tokens were erroneously reconstructed, and how many tokens are missing in the automatic reconstruction. Precision, recall and F1 measures can then be computed from these counts.

The baseline from which we started was reconstruction of only those tokens that have perfect correspondence, i.e., those tokens that appear identically in both the edited document and the output of the speech recognition system. Naturally, the precision of this approach lies at about 1.0, while the recall is quite low because for many regions, no reconstruction is created at all.

Further rules for handling specific phenomena like those described in section 4 were then added. Table 1 summarizes the improvements over the baseline approach. Evaluation was performed over a corpus of 1637 reports. There are some wrong reconstruction results even for the baseline mechanism: In a few rare cases, a token in the manual literal transcription does not match either of the token in the edited document or the output of the speech recognition. This might be due to an error of the transcriptionist, or due to a recognition error that found its way into the edited document.

We conclude from our evaluation of the reconstruction mechanism that good results can be achieved for certain phenomena. When the results are averaged over a big corpus, however, the relative improvement over the baseline approach is significant but less dramatic. Still, the results are encouraging in that there is a lot of potential left for handling further phenomena, thereby improving recall. It has been shown that reconstruction rules are typically of high precision, so that further improvements of the F1 measure are to be expected. Additionally, we expect that fine-tuning of existing reconstruction rules will further improve precision.

Acknowledgement

This research was carried out in the context of the SPARC project, a joint project by ÖFAI, the Institute for Signal Processing and Speech Communication at the Technical University of Graz, and Philips Speech Recognition Systems (PSRS). SPARC was funded by the FIT-IT program of the Austrian Federal Ministry for Transport, Innovation, and Technology. More information can be found at <http://www.sparc.or.at>.

Financial support for ÖFAI is provided by the Austrian Federal Ministry of Education, Science and Culture and by the Federal Ministry of Transport, Innovation and Technology.

References

1. Burgun, A. and Bodenreider, O.: 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"*, 2001:77-82, 2001.
2. Fellbaum C. (ed.): WordNet: An Electronic Lexical Database, MIT Press, 1998.
3. Huber M., Jancsary J., Klein A., Matiasek J., Trost H.: Mismatch interpretation by semantics-driven alignment. In *Proceedings of Konvens 2006*, 2006.
4. Levenshtein V.I.: Binary codes capable of correcting deletions, insertions, and reversals, *Doklady Akademii Nauk SSSR*, 163(4):845-848, 1965 (Russian). English translation in *Soviet Physics Doklady*, 10(8):707-710, 1966.
5. Pedersen T., Pakhomov S.V.S., Patwardhan S., and Chute Ch.G.: Measures of Semantic Similarity and Relatedness in the Biomedical Domain, *Journal of Biomedical Informatics*, 40(3), 288-299, June 2007.
6. Lindberg D.A.B, Humphreys B.L., McCray A.T.: The Unified Medical Language System, *Methods of Information in Medicine*, 32:281-291, 1993.
7. Petrik S., Kubin G.: Reconstructing medical dictations from automatically recognized and non-literal transcripts with phonetic similarity matching. *Proceedings of ICASSP-2007*, Honolulu, Hawaii, 2007.