# Lightweight Spoken Utterance Classification with CFG, tf-idf and Dynamic Programming

Manny Rayner(✉), Nikos Tsourakis, and Johanna Gerlach

TIM/FTI, University of Geneva, Geneva, Switzerland
{Emmanuel.Rayner,Nikolaos.Tsourakis,Johanna.Gerlach}@unige.ch

**Abstract.** We describe a simple spoken utterance classification method suitable for data-sparse domains which can be approximately described by CFG grammars. The central idea is to perform robust matching of CFG rules against output from a large-vocabulary recogniser, using a dynamic programming method which optimises the tf-idf score of the matched grammar string. We present results of experiments carried out on a substantial CFG-based medical speech translator and the publicly available Spoken CALL Shared Task. Robust utterance classification using the tf-idf method strongly outperforms plain CFG-based recognition for both domains. When comparing with Naive Bayes classifiers trained on data sampled from the CFG grammars, the tf-idf/dynamic programming method is much better on the complex speech translation domain, but worse on the simple Spoken CALL Shared Task domain.

**Keywords:** Speech recognition · Spoken utterance classification · Robustness · Context-free grammar · tf-idf · Medical applications

## 1 Overview

Spoken utterance classification is generally agreed to be an important problem, but published work to date has concentrated on a small number of scenarios, the most common of which are call routing and slot-filling applications like ATIS. It is in most cases assumed that there will be substantial amounts of training data available [5,7,8]. There are, however, many practically interesting types of application requiring spoken utterance classification which do not fit well into this picture. Our primary focus of interest here is fixed-phrase medical speech translators ("medical phraselators"). A medical phraselator contains on the order of thousands to tens of thousands of source-language utterances relevant to medical situations, each one paired with predefined translations in the target languages. The doctor speaks, and the app attempts to find the stored utterance closest

The work described here was funded by the Fondation privée des Hôpitaux universitaires de Genève and Unitec. We would like to thank Pierrette Bouillon for semantic annotation of the test data and many helpful suggestions, and Nuance Inc for generously making their software available to us for research purposes.

N. Camelin et al. (Eds.): SLSP 2017, LNAI 10583, pp. 143–154, 2017.
DOI: 10.1007/978-3-319-68456-7_12

to what they have said, showing it to the doctor to confirm that it has understood correctly; if the doctor approves the app's choice, it speaks a translation in the target language. The challenge is to make the matching process flexible and accurate, so that the users can express themselves reasonably freely and be correctly recognised most of the time. Since there are many semantic classes, and doctor time is scarce and hard to obtain, it is optimistic to expect more than small amounts of training data to be available until an advanced point in the project.

In the approach we describe here, we manually construct a CFG grammar which defines plausible variants for the questions, after which we robustly match spoken input to that CFG grammar. We have been surprised to find that a very simple matching method based on tf-idf indexing and dynamic programming gives quite good results. Although it seems plausible that a sophisticated modern deep learning method could achieve a lower error rate, the tf-idf method has definite advantages. It requires essentially no training data, is easy to implement, and is fast both at compile-time and at runtime. As noted, our main interest is in medical speech translation, but we also present results for the Spoken CALL Shared Task, an open dataset we recently have been involved in popularising.

The rest of the paper is organised as follows. Section 2 describes the two domains used. Section 3 describes the speech recognisers. Section 4 sketches experiments using Weka classifiers; these work well for the simple CALL domain, but much less well for the complex medical speech translation domain. The next two sections contain the main results of the paper: Sect. 5 describes the tf-idf/DP matching method, and Sect. 6 an evaluation on the two domains used. The final section concludes.

## 2 Domains Used

### 2.1 Medical Phraselators and the BabelDr Project

In the preceding section, we briefly outlined what we mean by a "medical phraselator". We have since 2015 been involved in a collaboration between the Geneva University Faculty of Translation and Interpreting and the Hôpitaux Universitaires de Genève (HUG), Geneva's largest hospital, whose goal is to produce a system of this general type. It is worth pointing out that medical phraselators have not been rendered obsolete by Google Translate (GT). A 2014 study [9] suggests that GT may mistranslate typical medical questions as much as 30% of the time; recent experiments carried out by our own group produce broadly similar results [3]. The problem is not so much the high error rate in itself as the fact that the only feedback given to the user, the recognition result, is very unreliable; GT often produces an incorrect translation after correct recognition. A phraselator, in contrast is explicitly designed to give dependable feedback.

The system we have developed, BabelDr (http://babeldr.unige.ch/; [4]), supports translation of medical examination questions from French into several languages, prioritising coverage relevant to Arabic- and Tigrinya-speaking migrants presenting at HUG's Accident & Emergency and migrant health departments.

The grammar has been written manually in a simple formalism based on Synchronous CFG [1]. The structure is “flat” and consists of a large set of top-level rules defining the various question patterns, together with more rules that define various kinds of phrase. The size of the generated coverage is of the order of tens or hundreds of millions of possible source-language sentences, mapping into of the order of thousands of semantic concepts. An example of a BabelDr rule is shown in Fig. 1. The `Source` lines define the actual CFG rule; the line marked `Target/french` is the backtranslation shown to the user at runtime. The backtranslations can also be accessed through a searchable help pane in the GUI.

```
Utterance
Source depuis combien d’heures \
       ($avez_vous | $ça_fait | $ressentez_vous) $mal_au_ventre
Source depuis combien d’heures $c_est_douloureux
Source ?(est-ce que) (ça | cela) fait combien d’heures que vous \
       (avez | ressentez | souffrez de)  $mal_au_ventre
Source combien d’heures (cela | ça) (fait | fait-il) que vous \
       (avez | ressentez | souffrez de) $mal_au_ventre
Source (il y a | ça fait) combien d’heures que vous \
       (avez | ressentez | souffrez de) $mal_au_ventre
Source combien d’heures (il y a | ça fait) que vous \
       (avez | ressentez | souffrez de) $mal_au_ventre
Target/french depuis combien d’heures avez-vous mal au ventre ?
EndUtterance
```

**Fig. 1.** BabelDr rule for the question *“Depuis combien d’heures avez-vous mal au ventre?”* (“For how many hours have you experienced stomach pain?”). We only show the source-language (French) side. Items starting with a dollar sign ($) are non-terminals.

In the initial version of the system, the grammar was compiled into a CFG-based language model and then into a recognition package that could be run on the Nuance Toolkit 10.2 engine [11]. This yielded a system which provided practically useful performance, but suffered from the usual problems associated with rule-based applications: performance was reasonably good for utterances inside grammar coverage but very poor on out-of-coverage ones, and it was too often difficult for the user to know where the dividing line went.

### 2.2 Data and CFG Grammars Used for Current Experiments

For the experiments carried out here, we had 965 utterances of recorded training data available. Test data was collected from medical students and doctors during December 2016 and January 2017, using a scenario in which the subjects used the earlier rule-based version of the system to communicate with simulated patients [3]. Data was logged and then transcribed and semantically annotated by the project member responsible for grammar development (not one of the authors). This produced a total of 827 utterances, of which 110 were annotated as being out of domain with respect to the grammar version used, i.e. not sufficiently closely associated with any of the semantic categories defined by the grammar. This left 717 in-domain utterances, containing 3794 words, which were used for the present experiments. Of these 717 in-domain utterances, 503 (70.2%) were inside grammar coverage.

The experiments described in this paper were performed using a version of the grammar chosen so that it predated the data collection exercise. The version used has a vocabulary of 2046 words, expands to about 45M possible strings, and defines 2187 possible semantic categories. Each semantic category has an associated backtranslation. We extracted the set of 2187 backtranslations, and used them as additional training data in ways described in more detail below.

### 2.3 The Spoken CALL Shared Task

The methods we describe here were motivated by the requirements of the BabelDr project, but in order to get some idea of their general applicability we also evaluated them on a second domain where we had suitable data readily available. The Spoken CALL Shared Task ([2]; https://regulus.unige.ch/spokencallsharedtask/) is a joint initiative by Geneva University, the University of Birmingham and Radboud University, whose goal has been to create an open challenge dataset in the area of prompt-response systems for speech-enabled Computer Assisted Language Learning ("spoken CALL"). Training data was released in July 2016, and test data in January 2017; the task received twenty submissions from nine different groups. Results were presented at the SLaTE workshop in August 2017 (http://www.slate2017.org/challenge.html).

The Shared Task dataset was collected using an online CALL app designed for Swiss German teens in their second or third year of learning English. Content was structured as a series of interactive dialogues, each one parametrized so that it could appear in many different variants, which allowed students to practice fluency and generative language skills. Like BabelDr, the CALL app used a Nuance recogniser with a language model derived from a CFG grammar, which associated each response with one or more prompts. This CFG grammar was made available as part of the Shared Task training data released. The grammar was not intended to be complete, and was only meant to be taken as providing a baseline.

A Shared Task item is a tuple consisting of the following elements: (a) a prompt; (b) a recorded audio file with the student's response; (c) a transcription; (d) a binary annotation (correct/incorrect) noting whether the audio file is a fully correct response to the prompt; (e) a binary annotation (correct/incorrect) noting whether the audio file is a semantically (but possibly not grammatically) correct response to the prompt. The last three fields are kept secret in the test data, and the task is to reproduce the (d) column. Shared Task data can easily be transformed into an utterance classification task by extracting the items where the response is marked as semantically correct. The semantic classification task is then to reconstruct the prompt given the audio file.

### 2.4 Data and CFG Grammars Used for Current Experiments

The training data used for the experiments was the 5222 utterance set released with the Spoken CALL Shared Task. This was available in two versions: as transcriptions, and as recognition results produced by the recogniser (cf. Sect. 3).

As test data, we used the portion of the Shared Task test data which was marked as semantically correct, transforming it as described above into data for an utterance classification task. The resulting dataset has 875 items containing 4630 words. 568 items (64.9%) were inside grammar coverage.

The grammar used was the one included in the Shared Task release. This has a vocabulary of 419 words, expands to about 45K possible strings, and defines 501 possible semantic categories.

## 3 Recognisers

In both domains, the baseline was thus defined by an annotated CFG grammar which also acted as a language model for a recogniser. The challenge was to make this baseline system robust to out-of-coverage utterances. We adopted an obvious strategy: use the available data to create a broad-coverage recogniser tuned to the domain and a robust classifier which associated recogniser output with the semantic classes defined by the CFG grammar. We start by describing the large-vocabulary recognisers, which were produced differently in the two domains:

**BabelDr.** We used the large vocabulary Nuance Transcription Engine, with an interpolated language model that combined the default language model with a model derived from the BabelDr training data.

**Shared Task.** We used the Kaldi recogniser developed by Mengjie Qian and colleagues at the University of Birmingham, the ASR data for which was publicly posted on the Shared Task site[1] under entry JJJ. The JJJ entry achieved the second best score on the Shared Task and is described in [10].

Table 1 presents basic performance results for the different recognisers when run on the test data, giving Word Error Rate (WER) and Sentence Error Rate (SER) for in-coverage, out-of-coverage and all data. For the grammar-based recogniser, we also present results for the portion of the test data where the confidence score is over the threshold. The threshold value of 0.65 was tuned on the Shared Task training data, also available from the Shared Task site. Performance was not sensitive to the exact setting, and threshold values between 0.60 and 0.70 gave similar results. Note that although the large-vocabulary recogniser strongly outperforms the grammar-based recogniser on the whole set, the converse relationship obtains on the "high confidence" subset of the data. As we will see later, this is why a hybrid system is able to outperform the plain robust system for both domains.

We now proceed to issues concerning semantic classification, which are the main subject of the paper.

[1] https://regulus.unige.ch/spokencallsharedtask, "Results" tab.

**Table 1.** Recogniser performance for BabelDr and Spoken CALL Shared Task domains on in-coverage, out-of-coverage and all data. The "%Data" column shows the proportion of the data for which the grammar-based recogniser is over the confidence threshold.

| Recogniser | %Data | IC | | OOC | | All | |
|---|---|---|---|---|---|---|---|
| | | WER | SER | WER | SER | WER | SER |
| *BabelDr* | | | | | | | |
| Grammar-based | (All) | 16.1 | 29.0 | 64.4 | 100.0 | 31.7 | 50.2 |
| Grammar-based (high confidence) | 38.9 | 3.4 | 13.5 | 39.5 | 100.0 | 6.8 | 19.7 |
| Large-vocabulary | (All) | 10.4 | 29.0 | 22.3 | 63.1 | 13.3 | 39.2 |
| *Spoken CALL Shared Task* | | | | | | | |
| Grammar-based | (All) | 17.2 | 28.2 | 53.4 | 99.3 | 30.0 | 53.1 |
| Grammar-based (high confidence) | 27.7 | 1.7 | 3.8 | 36.5 | 100.0 | 6.0 | 16.0 |
| Large-vocabulary | (All) | 7.9 | 21.1 | 18.6 | 52.8 | 11.7 | 32.2 |

## 4 Utterance Classification Using Weka

We began by testing performance, for the two domains used, of several popular classifiers supported by the Weka toolkit [6]. We report results for J48 decision trees, naive Bayes and SVM; other methods we tried gave clearly worse results. Our basic approach in all cases was to take labelled text data—sets of text strings representing utterances, each one paired with an associated semantic class—and extract unigram features, one for each word in the vocabulary.

For both domains we had a bit less than a thousand items of test data, in the form of labelled recognition results produced by recognisers. This data could reasonably be regarded as unseen for the purposes of the present experiments. The labelled training data we had available was fairly dissimilar for the two domains. For the Spoken CALL Shared Task, we had a substantial number (more than 5K) training examples, which were available both as transcriptions and as recognition results; for BabelDr, we had less than a thousand such examples. We did however have 2187 backtranslations, one for each semantic class. Since the backtranslations are both shown to the user after each turn and also available through the help system, users often imitated them, so we expected them to be a useful knowledge source.

Another important difference between the domains was in the grammars. The Spoken CALL Shared Task grammar was quite small; it was possible to expand it fully, giving about 45 thousand utterances, and use the whole grammar for training. The BabelDr grammar was much bigger, expanding to about 45 *million* utterances, and using the whole set was not feasible. Instead, we sampled the grammar randomly, creating 100 possible utterances from each rule. Table 2 summarises the domains and the available resources.

Table 3 presents the results. For the Spoken CALL Shared Task, the classification error was quite good even when training only on the transcriptions and recognition results, and improved further when the grammar data was added,

**Table 2.** Summary of available resources for the two domains

| | SharedTask | BabelDr |
|---|---|---|
| *Grammar* | | |
| #semantic categories | 501 | 2187 |
| #words of vocabulary | 419 | 2046 |
| #utterances in coverage | ∼45K | ∼45M |
| #utterances used for training | ∼45K | ∼220K |
| *Recorded training data* | | |
| #utterances training data | 5222 | 965 |
| *Backtranslations* | | |
| #backtranslations | – | 2187 |
| *Recorded test data* | | |
| #utterances test data | 875 | 717 |

reaching 11.8% for the best method. The figures for BabelDr, the domain we were actually interested in, were much less satisfactory, with a best error rate of 28.8%. On examining the results more closely, we thought one problem might be the fact that we were only using a small portion of the grammar. We consequently searched for a method which would let us use the whole grammar in some suitable form.

**Table 3.** Classification error rates using Weka methods on unseen spoken test data for the two domains. “J48” = J48 decision tree method. “NBayes” = Naive Bayes method. SVM training exceeded resource bounds for the BabelDr data.

| Training data | J48 | NBayes | SVM |
|---|---|---|---|
| *BabelDr* | | | |
| Backtranslations | 87.7 | 54.3 | – |
| Backtranslations + Transcriptions + Rec results | 55.2 | 38.4 | – |
| Backtranslations + Transcriptions + Rec results + Grammar | 34.1 | 28.8 | – |
| *Spoken CALL Shared Task* | | | |
| Transcriptions + Rec results | 16.2 | 15.9 | 13.9 |
| Transcriptions + Rec results + Grammar | 14.1 | 13.0 | 11.8 |

## 5 Utterance Classification Using tf-idf and Dynamic Programming

Attempting to find a way to use the whole grammar, rather than only a small part of it, two possible ideas suggested themselves to us. One was simply to try to find some kind of closest match between the string returned by the recogniser and a grammar rule. The other was to recast the problem as a type of document-indexing task, where the "documents" are the grammar rules. Specifically, we could use some version of the well-known tf-idf method [12] to find the rules which had high tf-idf scores with respect to the recogniser string; the tf-idf score basically measures the extent to which a word is a useful "keyword", i.e. occurs only in a small number of rules. In fact, it turned out to be easy to combine both ideas and split the problem into three parts. First, use tf-idf to find a small number of rules whose associated keywords match words in the recognition hypothesis produced by the recogniser; second, find the closest match between the recognition hypothesis and each rule in the shortlist produced by the first step; third, use information obtained from the matches to reorder the shortlist.

We approximate by treating the recogniser hypothesis as a bag of words rather than as an ordered string. This is an acceptable approximation for grammars like those considered here, where word-order is rarely important. It makes it possible to implement the matching process as a simple dynamic programming algorithm which recursively expands out the chosen grammar rule, chooses the best match for each piece, and combines the pieces. Since each grammar constituent only needs to be considered once, the process is very fast. In a little more detail, the currently implemented method is as follows:

1. At compile time, index words to associate them with the top-level rules in which they occur. Assign a word an idf score which is high if it occurs in few rules, low if it occurs in many rules. The simplest way to do this is to define the idf score for a word $W$ to be $1/f_W$, where $f_W$ is the number of top-level rules in which $W$ can occur; we may also smooth, use a logarithmic scale, etc. Call this mapping of words to rules and idf scores the *word to rules table.*
2. At compile time, associate each top-level rule with the closure of the set of non-top-level rules it may link to. Order these rules by the maximum depth at which they can occur. Call this mapping of rules to ordered lists of non-top-level rules the *rule to rule closure table.*
3. At runtime, the matcher is presented with a recognition hypothesis from the large-vocabulary recogniser. Use tf-idf to find the $n$ top-level rules with the best scores according to a naive scoring method which totals the tf-idf scores for all the words that are both used by the rule and also occur in the recognition hypothesis. This gives us a preliminary ordering of the rules.
4. For each rule in the $n$-best list created by the preceding step, perform a dynamic programming (DP) match against the recognition hypothesis, treating the hypothesis as a bag of words weighted by tf-idf scores. This DP match can be performed efficiently, since it is linear in the size of the grammar closure for the rule and logarithmic in the length of the input string. In more detail, the match proceeds as follows:
    (a) Begin by matching each phrasal rule in the rule closure list from (2), starting with the deepest ones, which are ordered to occur earliest in the list. The idea is that each rule will only be matched when all the non-terminals that can occur in it have already been matched. Associate each non-top-level rule with its best matching score and call the mapping of non-top-level rules to scores the *phrase score table.*

(b) To match a word in a CFG rule, check to see if it is in the input bag of words. If it is, add the tf-idf score from (1). If it isn't, add a fixed no-match penalty.
(c) To match a sequence $\langle P, Q\rangle$ in a CFG rule, match $P$ and $Q$ separately and assign a score which is the sum of the scores for $P$ and $Q$.
(d) To match an alternation $(P \mid Q)$ in a CFG rule, match $P$ and $Q$ separately and assign a score which is the larger of the scores for $P$ and $Q$.
(e) To match a nonterminal in a CFG rule, look up its best score in the phrase score table.
(f) At the end, add the fixed no-match penalty for each word in the input that has not been matched.

5. When all items in the $n$-best list have been matched, reorder them using the scores obtained in the previous step.

### 5.1 Refinements to the Basic Method

We tried a variety of tweaks to the basic method described above, including replacing the plain tf-idf scores with logarithmic scores and rescoring using the edit distance to the best grammar match measured in terms of the number of characters, the number of words, or the number of words weighted by the td-idf scores of the words affected. The only modification which had a positive effect on development set performance was one designed to address the problem of very unspecific rules, for example the rule associated with questions semantically equivalent to *Avez-vous mal?* ("Does it hurt?"). The problem with rules like these is that utterances matching them may fail to contain any word with a high tf-idf score, meaning that they cannot rise to the top of the $n$-best list. After some experimentation, the best solution found was to order the rules by minimum possible score at compile time, and at runtime always to add the $m$ potentially lowest-scoring rules. Based on the development set, we put $m = 3$.

## 6 Evaluation of the tf-idf/DP Method

We carried out a series of experiments to evaluate the tf-idf/DP method using the BabelDr and Spoken CALL Shared Task domains. For each domain, we compared four different versions of the system:

**Rule-based.** The pure rule-based version. Recognition is performed by the grammar-based language model, and semantic interpretation by the CFG.

**tf-idf.** A minimal robust version using the large-vocabulary recogniser together with semantic interpretation using only tf-idf. For this to be possible, we expanded the CFG rules to remove all the non-terminals and leave a flat grammar where each rule gave a single semantic result, and only used steps (1) and (3) from the sequence in Sect. 5.

**tf-idf/DP.** The full robust version, which combines the large-vocabulary recogniser and the complete semantic interpretation method from Sect. 5, including both tf-idf and DP matching.

**Hybrid.** A version which uses a simple method to combine the **Rule-based** and **tf-idf/DP** versions. For 1-best, the hypothesis is chosen from the rule-based system if the recogniser's confidence score is over a threshold, otherwise it is chosen from the robust system. For $n$-best $(n > 1)$, the hypotheses chosen are the 1-best result from the rule-based system and enough results from the robust system to make $n$ different hypotheses.

**Table 4.** 1-best and 2-best semantic classification error on unseen text and speech data for four different versions of the two systems, distinguishing between in-coverage, out-of-coverage and all input. Text input is transcribed speech input. "Rule-based" = pure rule-based system; "tf-idf" = robust system with only tf-idf; "tf-idf/DP" = full robust system; "hybrid" = hybrid system combining "rule-based" and "tf-idf/DP".

| Version | IC | | | | OOC | | | | All data | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Text | | Speech | | Text | | Speech | | Text | | Speech | |
| | 1-bst | 2-bst | 1-bst | 2-bst | 1-bst | 2-bst | 1-bst | 2-bst | 1-bst | 2-bst | 1-bst | 2-bst |
| *BabelDr* | | | | | | | | | | | | |
| Rule-based | (0) | (0) | 13.9 | 11.7 | (100) | (100) | 72.0 | 70.6 | 29.8 | 29.8 | 31.2 | 29.3 |
| tf-idf | 11.9 | 10.7 | 19.7 | 17.5 | 47.7 | 34.6 | 52.8 | 43.5 | 22.3 | 17.9 | 29.6 | 25.2 |
| tf-idf/DP | 1.2 | 0.0 | 8.5 | 6.2 | 43.5 | 28.5 | 48.1 | 39.3 | 13.8 | 8.6 | 20.4 | 16.0 |
| Hybrid | (0) | (0) | 6.4 | 1.6 | 43.5 | 28.5 | 48.1 | 38.8 | 13.8 | 8.6 | 18.8 | 12.7 |
| *Spoken CALL Shared Task* | | | | | | | | | | | | |
| Rule-based | (0) | (0) | 22.5 | 19.7 | (100) | (100) | 63.5 | 60.9 | 35.1 | 35.1 | 36.9 | 34.2 |
| tf-idf | 15.3 | 9.3 | 25.0 | 13.0 | 23.5 | 14.7 | 30.6 | 22.1 | 18.2 | 9.3 | 27.3 | 15.9 |
| tf-idf/DP | 1.8 | 0.5 | 11.8 | 7.2 | 20.2 | 14.0 | 30.9 | 21.2 | 8.2 | 5.3 | 18.5 | 12.2 |
| Hybrid | (0) | (0) | 9.5 | 6.0 | 20.2 | 14.0 | 30.6 | 22.5 | 8.2 | 5.3 | 16.9 | 11.8 |

Summary results for classification error on the test sets are presented in Table 4, which shows 1-best and 2-best error rates for text and speech input, and Table 5, which breaks down results for the robust versions as a function of the number of word errors in the large-vocabulary recogniser's output. Rather surprisingly, the first impression is that performance on the two domains is reasonably similar. Looking first at Table 3, we see that WER over the whole test set is 12–13% for the large-vocabulary recogniser. For the grammar-based recogniser it is about 30% for the whole set and about 6–7% for the subset where the confidence score is over the threshold.

Turning next to Table 4, we see that 1-best semantic classification error on the whole set using the pure rule-based system is about 30–40% for spoken input. This is reduced to 17–19% for the hybrid version. 2-best error reduces from 30–35% to about 12–13%. The relative improvement in 1-best error is 40% for BabelDr and 54% for Shared Task; for 2-best error, it is 57% for BabelDr and 65% for Shared Task. The larger improvement in the Shared Task system is consistent with the fact that its CFG grammar represents a much smaller development effort and is less carefully constructed. Comparing the lines for

**Table 5.** 1-best and 2-best semantic classification error as a function of number of word errors. #Errs = number of word errors in 1-best speech recognition hypothesis; #Sents = number of examples with given number of word errors

| #Errs | #Sents | tf-idf | | tf-idf/DP | | Hybrid | |
|---|---|---|---|---|---|---|---|
| | | 1-bst | 2-bst | 1-bst | 2-bst | 1-bst | 2-bst |
| *BabelDr* | | | | | | | |
| 0 | 453 | 18.3 | 15.0 | 7.5 | 4.0 | 7.3 | 5.5 |
| 1 | 121 | 41.3 | 37.2 | 30.6 | 28.2 | 27.3 | 19.8 |
| 2 | 75 | 54.7 | 44.0 | 54.7 | 42.7 | 49.3 | 30.7 |
| >2 | 68 | 55.9 | 51.5 | 50.0 | 45.6 | 47.1 | 27.9 |
| *Spoken CALL Shared Task* | | | | | | | |
| 0 | 593 | 15.3 | 7.1 | 5.6 | 2.0 | 5.9 | 3.5 |
| 1 | 117 | 39.3 | 30.8 | 36.8 | 28.2 | 32.5 | 21.4 |
| 2 | 110 | 56.4 | 26.4 | 49.1 | 29.1 | 40.0 | 30.9 |
| >2 | 55 | 72.7 | 58.2 | 58.2 | 52.7 | 56.4 | 41.8 |

plain tf-idf, tf-idf/DP and hybrid, we see that inclusion of the DP matching step makes a large difference, particularly on in-coverage data, and hybrid improves non-trivially on tf-idf/DP.

Finally, Table 5 measures robustness to recognition errors. The hybrid system achieves a 1-best classification error of 6–7% on utterances which are correctly recognised, falling to about 30% on utterances with one recognition error, 40–45% on utterances with two recognition errors, and 50–55% on utterances with more than two recognition errors. The contribution of DP matching is most important on correctly recognised utterances. The largest differences occur on text input, which we included to give a baseline approximating perfect recognition. The higher error rate on BabelDr data (13.8% versus 8.2%) probably reflects the more challenging nature of the domain.

The dynamic programming matching method is fast both at compile-time and at runtime. Running on a 2.5 GHz Intel laptop, compilation of the tables required by the tf-idf/DP method requires less than a minute for each domain. Average processing time at runtime is about 40 ms/utterance.

## 7 Conclusions and Further Directions

We have presented a simple spoken utterance classification method suitable for domains which have little training data and can be approximately described by CFG grammars, and evaluated it on two such domains. Compared to plain CFG-based classification, the method reduces 1-best error on spoken input by over a third on the well-tuned BabelDr domain and over a half on the poorly-tuned Shared Task domain. We find these results encouraging, not least because the methods so far implemented can very likely be improved. Two obvious things to try next are introducing a better treatment of OOV words, which at the moment are uniformly counted as skipped, and simply tuning the recogniser more.

Our practical goal in this project has been to improve the BabelDr system. From a theoretical point of view, however, the most interesting finding has been the contrast between the mainstream Weka methods and tf-idf/DP. On the small Shared Task domain, the Weka methods strongly outperform tf-idf/DP, with the Naive Bayes method achieving a classification error of 13.0% as compared to the "hybrid" method's 16.9%. On the much more challenging BabelDr domain, however, the pattern is reversed. Naive Bayes scores 28.8%—only slightly better than the baseline CFG—while "hybrid" reduces the error to 18.8%. As noted, we think the poor performance of the Weka methods may reflect the inadequacy of creating training data by random sampling from the grammar, and it is possible that some more intelligent sampling method may allow us to address the problem. We are currently investigating this.

## References

1. Aho, A.V., Ullman, J.D.: Properties of syntax directed translations. J. Comput. Syst. Sci. **3**(3), 319–334 (1969)
2. Baur, C., Chua, C., Gerlach, J., Rayner, E., Russell, M., Strik, H., Wei, X.: Overview of the 2017 spoken CALL shared task. In: Proceedings of the Seventh SLaTE Workshop, Stockholm, Sweden (2017)
3. Bouillon, P., Gerlach, J., Spechbach, H., Tsourakis, N., Halimi, S.: BabelDr vs Google Translate: a user study at Geneva University Hospitals (HUG). In: Proceedings of the 20th Conference of the European Association for Machine Translation (EAMT), Prague, Czech Republic (2017)
4. Bouillon, P., Spechbach, H.: BabelDr: a web platform for rapid construction of phrasebook-style medical speech translation applications. In: Proceedings of EAMT 2016, Vilnius, Latvia (2016)
5. Hakkani-Tür, D., Béchet, F., Riccardi, G., Tur, G.: Beyond ASR 1-best: using word confusion networks in spoken language understanding. Comput. Speech Lang. **20**(4), 495–514 (2006)
6. Holmes, G., Donkin, A., Witten, I.H.: Weka: A machine learning workbench. In: Proceedings of the Second Australian and New Zealand Conference on Intelligent Information Systems, pp. 357–361. IEEE (1994)
7. Kuo, H.K.J., Lee, C.H., Zitouni, I., Fosler-Lussier, E., Ammicht, E.: Discriminative training for call classification and routing. Training **8**, 9 (2002)
8. Mesnil, G., He, X., Deng, L., Bengio, Y.: Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In: Interspeech, pp. 3771–3775 (2013)
9. Patil, S., Davies, P.: Use of Google Translate in medical communication: evaluation of accuracy. BMJ **349**, g7392 (2014)
10. Qian, M., Wei, X., Jancovic, P., Russell, M.: The University of Birmingham 2017 SLaTE CALL shared task systems. In: Proceedings of the Seventh SLaTE Workshop, Stockholm, Sweden (2017)
11. Rayner, M., Bouillon, P., Ebling, S., Strasly, I., Tsourakis, N.: A framework for rapid development of limited-domain speech-to-sign phrasal translators. In: Proceedings of the workshop on Future and Emerging Trends in Language Technology, Sevilla, Spain (2015)
12. Sparck Jones, K.: A statistical interpretation of term specificity and its application in retrieval. J. Doc. **28**(1), 11–21 (1972)