Hybrid methods for ICD-10 coding of death certificates

Pierre Zweigenbaum
LIMSI, CNRS,
Université Paris-Saclay,
F-91405 Orsay, France
pz@limsi.fr

Thomas Lavergne
LIMSI, CNRS, Univ. Paris-Sud
Université Paris-Saclay,
F-91405 Orsay, France
lavergne@limsi.fr

Abstract

ICD-10 coding of death certificates has received renewed attention recently with the organization of the CLEF eHealth 2016 clinical information extraction task (CLEF eHealth 2016 Task 2). This task has been addressed either with dictionary projection methods or with supervised machine learning methods, but none of the participants have tried to design hybrid methods to process these data. The goal of the present paper is to explore such hybrid methods. It proposes several hybrid methods which outperform both plain dictionary projection and supervised machine learning on the training set. On the official test set, it obtains an F-measure of 0.8586 which is 1pt above the best published results so far on this corpus ($p < 10^{-4}$). Moreover, it does so with no manual dictionary tuning, and thus has potential for generalization to other languages with little effort.

1 Introduction

Biomedical information processing crucially relies on a normalized representation of medical information in the form or standardized terminologies and ontologies, be it for clinical care (SNOMED, LOINC), for public health statistics and health management (International Classification of Diseases) or for literature search (MeSH). Automatically generating such a normalized representation from naturally occurring sources such as text is therefore a long-studied goal (Wingert et al., 1989). Basically, it consists in deciding which concepts in the target representation (e.g., signs and symptom concepts in SNOMED CT, or disease classes in the ICD-10 classification) best represent the contents of a given text (e.g., a patient discharge summary). It can be decomposed into the detection of text mentions of biomedical concepts of the suitable types (entity recognition) and the determination of the target concepts (concept normalization) which best represent the text mentions in the context of the source text and the given use case. The state of the art of biomedical entity recognition and biomedical concept normalization has been established and published in a number of shared tasks which addressed clinical texts (Pestian et al., 2007; Uzuner et al., 2007; Uzuner et al., 2011; Suominen et al., 2013), biomedical literature (Kim et al., 2011; Nédellec et al., 2015), sometimes in multiple languages (Suominen et al., 2013; Névéol et al., 2016).

This paper focuses on ICD-10 coding. ICD coding has been studied in the past (e.g., as early as (Wingert et al., 1989)), but only recently has a large dataset been released for ICD-10 coding of death certificates (Névéol et al., 2016). In that context, Névéol et al. (2016) mention that participants in the CLEF eHealth 2016 ICD-10 coding task either used dictionary-based methods or supervised machine learning methods, and that none tried hybrid methods. The goal of this paper is to explore this direction. Our contributions are the following:

- We explore hybrid methods for ICD-10 coding which combine dictionary projection and supervised machine learning.
- We show that simple hybrid combinations with union and intersection yield improved results.
- We propose methods which improve the precision of dictionary projection, including hybrid 'calibration' methods.
- The methods which fare best on the training corpus, when applied to the test corpus, are

Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis (LOUHI), pages 96–105,
Austin, TX, November 5, 2016.

on par with the best published results on this corpus, with no manual dictionary tuning, and have thus potential for generalization to other languages with little effort.

In the remainder of the paper, we report the methods used by the best-performing participants in the CLEF eHealth 2016 shared task (Section 2), present the methods we explored and the data on which we applied them (Section 3), the results we obtained on the development and test data (Section 4), discuss them (Section 5) and conclude (Section 6).

2 Related Work

When producing normalized concepts from medical texts, most methods use dictionary-based lexical matching or supervised machine-learning. Most dictionary-based methods use the UMLS (Bodenreider, 2004) or one of its included vocabularies, such as the ICD-10 classification. MetaMap (Aronson and Lang, 2010) is the most used system for English: it takes advantage of the term variants present in the UMLS MetaThesaurus and of the morphological knowledge provided by the UMLS Specialist Lexicon. Knowledge-lean methods based on approximate dictionary look-up have also been proposed (Zhou et al., 2006).

Some studies have addressed the ICD-10 coding of death certificates. Koopman et al. (2015a) classified Australian death certificates into 3-digit ICD-10 codes such as *E10* with SVM classifiers based on n-grams and SNOMED CT concepts, and with rules. They also trained SVM classifiers (Koopman et al., 2015b) to find ICD-10 diagnostic codes for death certificates. In contrast to the above-mentioned CLEF eHealth shared task, they only addressed cancer-related certificates: they set-up a first-level classifier to detect cases of cancer then a second-level classifier to refine it into a specific type. Another difference from CLEF eHealth is that they remained at the level of 3-digit ICD-10 codes (e.g., *C00*, *C97*) instead of the full 4-digit level usually required for ICD-10 coding (e.g., *C90.9*). Another important difference is that they targeted the underlying cause of death, i.e., one diagnosis per death certificate, whereas the CLEF eHealth task requires to determine all the diagnoses mentioned in each statement of a given death certificate.

Two ICD coding shared task were organized so far. The Computational Medicine Center (CMC) challenge (Pestian et al., 2007) targeted ICD-9-CM disease coding from outpatient chest x-ray and renal procedures, whose clinical history and impression sections provide most support for coding. The dataset contained 978 documents for training and 976 documents for testing. It targeted a small subset of 45 ICD-9-CM codes, designed in such a way that every one of the 94 distinct combination of codes present in the test set were seen in the training set. The best system was based on a supervised classifier (a Decision Tree) and obtained an F-measure of 0.89 on the test set.

The CLEF eHealth 2016 ICD-10 coding task (Névéol et al., 2016) provided a dataset which consisted of death certificates in French. These death certificates were provided by CépiDc, the WHO collaborating center which manages ICD-10 coding of death certificates in France. We reproduce the corpus statistics from the task organizers' paper in Table 1. The task was defined at the level of each statement (line) in a death certificate: one statement could be associated with 0, 1 or more ICD-10 codes which represent causes of death at various levels in the causal chain which led to the death. Statements have a length which varies from 1 to 30 words, with outliers at 120 words and the most frequent length at 2 tokens. They are thus much shorter than the CMC challenge texts.

	Training (2006–2012)	Test (2013)
Documents	65,844	27,850
Lines	195,204	80,899
Tokens	1,176,994	496,649
Total ICD codes	266,808	110,869
Unique ICD codes	3,233	2,363

Table 1: The CépiDC French Death Certificates Corpus (from Névéol et al.).

The full dataset contained death certificates from 2006 to 2013. In a natural use case, death certificates of former years have already been coded and are available as examples to code new death certificates. Therefore the test corpus contained certificates of year 2013, whereas the training corpus contained certificates of years 2006–2012. There was

therefore no guarantee that a code needed in 2013 had been used in 2006-2012: a posteriori analysis reveals that 224 of the 2,363 unique codes used in 2013 were not used in 2006–2012. Besides, as can be seen in Table 1, the size of the corpus is much larger than that of the CCMC challenge, as well as the number of target codes.

Table 2 shows examples statements from the dataset; we provided English translations for the reader's convenience.

Statement + English gloss	Codes
surinfection superinfection	B99
insuffisance respiratoire aiguë acute respiratory failure	J960
arrêt cardio-respiratoire hypoxémique hypoxaemic cardio-respiratory arrest	R092, R090
Hypertrophie ventriculaire gauche concentrique d'étiologie indéterminée Concentric left ventricular hypertrophy of unknown origin	I517
Epilepsie séquellaire à AVC sylvien droit, AC/FA chronique, insuffisance cardiaque congestive, insuffisance rénale, atélectasie pulmonaire Sequelar epilepsy with right sylvian stroke, chronic atrial fibrillation/cardiac arrhythmia, congestive heart failure, renal failure, pulmonary atelectasis	J981, I500, G409, I48, I64, N19

Table 2: Statement examples with their associated ICD-10 codes, with English glosses. Code order does not necessarily align with text order.

CLEF participants were also provided with dictionaries created by CépiDc for their own use. Each dictionary included (term, ICD-10 code, related code 1, related code 2) quadruplets. We did not use the two 'related codes', hence only consider (term, ICD-10 code) pairs in the remainder of this paper. Four dictionaries were provided: one used over the years 2006–2010 (157,001 lines), one for 2011 (156,937 lines), one for 2012 (158,163 lines), and one for 2013 (144,905 lines). These dictionaries reflect changes in coding practice over the years, either caused by changes in international ICD contents or coding rules, or by newly encountered expressions which were not covered in previous years, or by improvements in CépiDc's dictionary management.

The top two systems at the CLEF eHealth ICD-10 coding task used two different methods.

Van Mulligen et al. (2016) relied on ICD dictionaries built from the shared task data. Their baseline dictionary used the term-code associations seen in the shared task training set, and their expanded dictionary also used the above-mentioned CépiDc dictionaries. Various filters were applied to these dictionaries, based on the ambiguity of the term-code associations. Their dictionary projection method used the Solr information-retrieval system to cope with the large number of entries in the lexicon efficiently. After measuring its performance on the training corpus, they post-processed their system output to block term-code associations with a precision on the training set lower than a given threshold selected by optimizing F-measure on the training set. They obtained the top precision, recall, and F-measure published so far on this dataset: P=0.886, R=0.813, F=0.848 in their top run using the expanded dictionary, or P=0.890, R=0.803, F=0.844 in their second run using the baseline dictionary.

Instead of trying to spot occurrences of known terms or variants in the input statements and then normalize them to ICD codes, Dermouche et al. (2016) addressed the task as a text classification problem: given a short text, compute a class, here an ICD-10 code. They used a supervised machine learning method (SVM) with bags of words after text preprocessing. They also tested transformations of the obtained vector space representation with topic models. The precision of their best submitted run (P=0.882) was very close to the that of the top system but their recall and F-measure were lower (P=0.882, R=0.655, F=0.752). The probable reason for their lower recall was that they produced one code per statement (mono-label classification), whereas given the data in Table 1, we can compute that there was an average of 1.37 codes per source statement both in the training corpus and in the test corpus. If a similar method could address multi-label classification and scale its recall linearly, it would reach a recall of $0.655 \times 1.37 = 0.897$, even higher than the dictionary projection method, which naturally performs multi-label classification.

As mentioned in the introduction, Névéol et al. (2016) observed that no participant in the CLEF eHealth 2016 ICD-10 coding task tried hybrid methods which would combine dictionary projection and supervised machine learning. Exploring this direction is the goal of this paper.

3 Methods

We set up a simple dictionary projection method and a supervised machine learning method, then designed hybrid methods based on one or both of them.

We first processed each statement as follows: conversion to lower case, tokenization (with an NLTK regular expression), stop word removal (French NLTK); diacritic removal (Unicode 'NFD' normalization), correction of some spelling errors based on the words present in the training corpus and in the CépiDc dictionaries, stemming (Snowball French stemmer).

3.1 Dictionary projection

Dictionary projection relies on the expressions present in a dictionary to spot mentions of concepts in a text. We pre-processed the CépiDc dictionaries in the same way as the death certificate statements: as a result, each dictionary entry links a sequence of normalized tokens to one or more ICD codes. For term matching efficiency, each dictionary was stored as a Trie. Given a dictionary, an input sequence of tokens is processed as follows. The input sequence of tokens is scanned for the first match. In case of multiple matches, the longest match is retained. After a match, scanning resumes right after the end of the match. The output of the process is a (possibly empty) list of matched dictionary entries together with their positions in the input sequence.

No processing of negations was performed because statements are very short and negations are infrequent. For instance, only 82 occurrences of the negation *pas* (*no/not*) were found in the training corpus (i.e., in 0.04% of the statements), and 240 occurrences of the negation *sans* (*without*) (0.12%).

A dictionary entry may lead to `0:n` codes. Depending on how the dictionary was built, the same code may have been recorded multiple times: this number of times is recorded in the dictionary. We have tested the following selection strategies in case of multiple outputs for a given entry:

all All codes are returned.

best The most frequently recorded code is returned. In case of a tie, a random choice is performed.

boiu (Best Only If Unambiguous): The most frequently recorded code is returned only if there is no tie, else no result is returned.

Dictionary projection can use any available dictionary which links terms to ICD codes. Here we tested only those provided by CépiDc to the CLEF eHealth participants: the use of other dictionaries which could be built for instance from the training corpus, from the ICD-10 terms themselves, or from the UMLS, is left for future work.

3.2 Supervised classification

Supervised classification is not the focus of this paper, therefore we only present here our best current model. It uses a linear SVM classifier and the following method and features:

- Linear SVM (scikit-learn's LinearSVC with default parameters, which relies on liblinear)
- Tokens (t), obtained after the above-mentioned pre-processing step. We also tested token n-grams up to 5, but this did not improve the results.
- Character trigrams (*c3*): spelling errors are frequent in the certificates; representing a statement by its overlapping character trigrams provides a degree of robustness to spelling errors.
- Coding Year (y): coding rules change over the years, and the same statement seen at two different dates may be coded differently because of such changes. Therefore we found it useful to include 2×9 features instantiated for $y \in [2006 \ldots 2014]$: '$> y$' or '$\leq y$' depending on the value of the Coding Year (e.g., a statement of 2011 will have '$>$2006', ... '$>$2010', '$\leq$2011', ... '$\leq$2014'.

This supervised classifier uses no information on ICD terms or codes other than that present in its training corpus.

3.3 Union and intersection of classifiers

The union of the outputs of two classifiers is a very simple method to combine them. It is useful when the individual classifiers lack recall, and preferably have a high enough precision. The ideal situation occurs when individual classifiers output different correct predictions (in which case the resulting recall will be higher than the best recall of the individual classifiers) and when the individual classifiers make errors on the same inputs (in which case the resulting number of false positives will be lower than the sum of the individual false positives).

Conversely, the intersection of two classifiers is a possible method to increase their precision. A high-precision classifier is useful for pre-annotation. In the actual coding process at CépiDc, human coders spend a sizable part of their time assigning codes which are easy to determine. Pre-annotating these codes with a reliable, high-precision system before presenting death certificates to human coders would enable them to browse through these pre-assigned codes quickly. This would save human coding time which could be reassigned to solving more difficult cases.

3.4 Calibration

A prediction method, for instance dictionary projection, can be 'calibrated' by training a classifier to detect its errors. Calibration takes into account the distribution of codes and of prediction success in the training split, thereby adding data-driven knowledge to the application of the expert-produced dictionary. It automatically spots the main deficiencies of the dictionary projection and blocks them. In this respect, it is closely related to the error analysis process which a human expert performs when applying their dictionary to a new dataset: error spotting, then correction. In the human process, correction can take the form of simple post-processing rules which filter out output codes known to be often erroneous. It can also come from data-driven tuning of the dictionary by measuring the performance of its entries on the training corpus and selecting an appropriate threshold to prune low-performance entries, as in (Van Mulligen et al., 2016). This is exactly what is performed automatically by the classifier we train.

We trained a classifier with the following conditions:

- Classifier: Linear SVM (scikit-learn's LinearSVC with default parameters).
- Features: individual code predicted by the CépiDc dictionary (see below Section 3.5), prefixed by *code:* (e.g., *code:R068*); we also tested the addition of the statement tokens (obtained by the same process as described above).
- Classes: True (meaning the predicted code is correct) / False (meaning it is incorrect).
- Training: our training split (see below: 185k statements) for development, the full training corpus for testing.

When testing, the trained classifier was applied to each individual code predicted by the dictionary projection. If the classifier's output was the False class, the predicted code was removed from the dictionary projection output.

3.5 Data

We used the CépiDc data provided by the CLEF eHealth 2016 clinical information extraction task (CLEF eHealth 2016 Task 2) to the challenge participants (Névéol et al., 2016). The statistics of the training and test corpora are described in Section 2. To emulate the test conditions in our development phase, we also split the training corpus based on the dates of the certificates: the last 10,000 statements (1141 unique codes) made up our test split, while the first 185,204 statements (13,300 codes, 3,200 unique) constituted our training split. Only 11 codes were present in the test split but absent from the training split.

Python 3.5.2 was used for the programs, with scikit-learn 0.17.1, within Anaconda 4.0.0.

3.6 Experimental protocol and evaluation

Teams were allowed to submit up to three runs to the task. In the present work, we emulated the same situation and selected three methods to run on the test corpus based on their F-measures in our experiments on the training corpus. This prevented us from biasing the final results by tuning them on the test corpus. To apply these methods to the test corpus,

we retrained them on the full training corpus with a more recent dictionary:

- The supervised classifier (Linear SVM, *tc3y*) was trained on the full training corpus.
- Dictionary projection methods used the 2012 dictionary instead of the 2011 dictionary.
- Dictionary projection was calibrated on the full training corpus.
- Supervised classifier and calibrated dictionary projection were applied to the test corpus.
- The union of their results was computed and used as final predictions.

Precision, recall and F-measure were computed for each experiment, by our own programs for convenience during development; when applied to the test corpus, they were computed with the official scoring program provided to the CLEF eHealth participants.

4 Results

4.1 Development: results on the test split of the training corpus

The SVM classifier with tokens, character trigrams, and year coded (henceforth *tc3y*), was trained on our training split and applied to our test split, on which it obtained P=0.9010, R=0.6774, and F=0.7734.

We tested the four dictionaries and our three dictionary output selection methods on our test split. Table 3 shows that the 2011 dictionary obtains the best precision, recall and F-measure, closely followed by the 2012 dictionary. As could be expected, the *all* method always produced the highest recall, whereas the *boiu* method always produced the highest precision. *boiu* also obtained the highest F-measure. The top F-measure was thus obtained with the 2011 dictionary and *boiu*, at P=0.8048, R=0.6475 and F=0.7176. We therefore retained the 2011 dictionary for further experiments on our test split (2012 data). We also assumed that following the same pattern for the official test data, dated in 2013, the 2012 dictionary should be most suitable. As a safety check, we tested the 2012 dictionary on our test split in the same conditions as the 2011 dictionary, and observed that it obtained similar results—slightly inferior, by a maximum of 0.1 pt P, R or F.

Dict	Sel	# Sys	TP	P	R	F
2006	boiu	10720	8470	0.7901	0.6368	0.7052
2006	best	12977	9117	0.7026	0.6855	0.6939
2006	all	18458	10133	0.5490	0.7619	0.6381
2011	boiu	10701	8612	**0.8048**	0.6475	**0.7176**
2011	best	12978	9335	0.7193	0.7019	0.7105
2011	all	18722	10491	0.5604	**0.7888**	0.6552
2012	boiu	10580	8485	**0.8020**	0.6380	**0.7106**
2012	best	12970	9276	0.7152	0.6974	0.7062
2012	all	18520	10469	0.5653	**0.7871**	0.6580
2013	boiu	10550	8106	0.7683	0.6095	0.6797
2013	best	13095	8951	0.6835	0.6730	0.6782
2013	all	19285	9956	0.5163	0.7486	0.6111

Table 3: Dictionary experiments on our test split: CépiDc dictionaries (Dict), 10,000 statements, 13,300 codes: all statements date from year 2012. Sel = Selection method: boiu = best only if unambiguous, best = most frequent code (random choice in case of tie), all = all codes. # Sys = number of system-predicted codes. TP = true positives. P = precision, R = recall, F = F-measure.

Table 4 shows the 2011 dictionary results without (–) and with (*c, c-t*) calibration. Calibration based only on the dictionary-proposed code (*Cal=c*) boosts precision by 12pt (*boiu*) to 33pt (*all*) and F-measure by 2.6pt (*boiu*) to 14pt (*all*), while only reducing recall by 2.5pt (*boiu*) to 6pt (*all*). Additionally taking into account the tokens of the coded statement in calibration (*Cal=c-t*) adds another 1.7pt (*boiu* or *all*) to 1.9pt (*best*) to precision and 0.25pt (*boiu*) to 0.6pt (*all*) to F-measure, with a decrease of recall by 0.15pt (*all*) to 0.4pt (*boiu* or *best*). Altogether, calibration is therefore highly efficient on our test split to increase precision and F-measure. The highest precision is obtained with *boiu, c-t* while the highest F-measure is obtained with *all, c-t*.

We performed the union and the intersection of the outputs of the SVM supervised classifier and of the dictionary projection. The results are reported in Table 5.

Union with the non-calibrated dictionary projection decreased its precision only by 1pt (*boiu*) or even increased it by 1 or 2pt (*best, all*) because the supervised classifier had a much higher precision, at

Sel	Cal	# Sys	TP	P	R	F
boiu	–	10701	8612	0.8048	**0.6475**	0.7176
boiu	c	8971	8276	0.9225	0.6223	0.7432
boiu	c-t	8749	8221	**0.9397**	0.6181	**0.7457**
best	–	12978	9335	0.7193	**0.7019**	0.7105
best	c	9769	8823	0.9032	0.6634	0.7649
best	c-t	9514	8773	**0.9221**	0.6596	**0.7691**
all	–	18722	10491	0.5604	**0.7888**	0.6552
all	c	10809	9631	0.8910	0.7241	0.7990
all	c-t	10585	9610	**0.9079**	0.7226	**0.8047**

Table 4: 2011 dictionary calibration experiments on our test split. Cal = calibration: – (none), c (dictionary code), t (source tokens).

the same time boosting recall by 12 to 20pt, reaching a maximum of 0.9048. Union with the calibrated dictionary projection decreased its precision by at most 5pt (*boiu, c-t*), maintaining a very reasonable P=0.86–0.89. Recall was boosted by 14 to 19pt, leading to a record F-measure of 0.8666.

Again, *all* obtained the highest recall and also the highest F-measure, achieving records of R=0.8661 (–) and F=0.8666 (*c-t*, both with quite balanced P, R, F. The *all c-t* combination was thus a natural candidate to run on the official test corpus.

With intersection, the obtained precision gained 3.5pt over the best so far, reaching 0.96–0.97, while losing 16–19pt of recall at 0.46–0.54 compared to the calibrated dictionary projection. Here again, *boiu* obtained the highest precisions with the top at 0.9749 (*c-t*). Intersection yields a 58% reduction of the best error rate so far from 6% to 2.5%. With such a low error rate, pre-annotation becomes viable and would cater for not far from one half of the number of codes to produce (R=0.4620). For information, we added this precision-oriented configuration (*boiu c-t*) to the three F-measure-oriented configurations to be run on the official test corpus.

4.2 Results on the test corpus

The best F-measure on the training corpus was obtained by the union of the SVM classifier and the *all* dictionary projection calibrated with token features (*all-c-t*), therefore we selected this method as our Run 1. We wanted to diversify our tests, therefore also selected two more precise runs: (ii) the union of the SVM classifier and the *boiu* dictionary

Sel	Cal	# Sys	TP	P	R	F
svm (linear) tc3y			9010	0.9010	0.6774	0.7734
Union						
boiu	–	14188	11303	0.7967	0.8498	0.8224
boiu	c	12670	11153	0.8803	0.8386	0.8589
boiu	c-t	12447	11087	0.8907	0.8336	0.8612
best	–	15894	11566	0.7277	0.8696	0.7924
best	c	13017	11313	0.8691	0.8506	0.8597
best	c-t	12719	11224	0.8825	0.8439	0.8628
all	–	20836	12034	0.5776	**0.9048**	0.7051
all	c	13414	11519	0.8587	0.8661	0.8624
all	c-t	13142	11457	0.8718	0.8614	**0.8666**
Intersection						
boiu	–	6293	6128	**0.9738**	0.4608	0.6255
boiu	c	6291	6127	**0.9739**	0.4607	0.6255
boiu	c-t	6302	6144	**0.9749**	0.4620	0.6269
best	–	7084	6779	0.9569	0.5097	0.6651
best	c	6752	6520	0.9656	0.4902	0.6503
best	c-t	6795	6559	0.9653	0.4932	0.6528
all	–	7886	7467	0.9469	**0.5614**	**0.7049**
all	c	7395	7122	0.9631	0.5355	0.6883
all	c-t	7443	7163	0.9624	0.5386	0.6906

Table 5: 2011 dictionary experiments on our test split: Union and intersection of Linear SVM and dictionary results.

projection calibrated with token features (*boiu-c-t*), and (iii) the union of the SVM classifier and the *boiu* dictionary projection calibrated with no extra features (*boiu-c*). We applied these methods to the test corpus in the manner presented above.

The results obtained on the official test corpus are very close to those on our test split of the training corpus: there is a constant difference of only –0.8pt in F-measure, and a similarly small decrease of less than 1pt in precision and recall for the three runs. This shows that the tested methods do not overfit the training corpus. As a consequence, the order of results on the test corpus reproduces that of the test split: highest F-measure and recall for *u(lsvcd(tc3y),d2012-all-c-t)*, highest precision for *u(lsvcd(tc3y),d2012-boiu-c-t)*.

The F-measures of the three selected runs exceed that of the best CLEF eHealth participant (P=0.886, R=0.813, F=0.848) by 0.3 to 1pt and their recalls do so by 1 to 4pt, whereas the precisions of these runs are below the best CLEF precision (P=0.890) by 0.6 to 2.5pt. Because of the large size of the test corpus, all of the differences from the best CLEF run (see Table 6) are significant (tested with ap-

Method	P	R	F
svm (tc3y)	0.8938	0.6645	0.7623
u(svm,d-all-c-t)	0.8656^{-4}	$\mathbf{0.8517}^{-4}$	$\mathbf{0.8586}^{-4}$
u(svm,d-boiu-c-t)	**0.8840**	0.8242^{-4}	0.8531^{-4}
u(svm,d-boiu-c)	0.8751^{-4}	0.8282^{-4}	0.8510^{-3}
i(svm,d-boiu-c-t)	**0.9703**	0.4500	0.6148

Table 6: Tests on the official test corpus. Evaluation with the official program. u(a,b) = union(a,b). i(a,b) = intersection(a,b). svm is a linear SVM with features tc3y. d- = dictionary (2012). In Union results, superscripts represent the power of the p-value of significance testing for the difference with the best published result so far (Van Mulligen et al., 2016): $-3 = p < 10^{-3}$, $-4 = p < 10^{-4}$. Note that the values of P are higher in (Van Mulligen et al., 2016) (the difference is significant in two cases out of three) whereas the values of R and F are better in the present Union results (differences are always significant).

proximate randomization with 10,000 permutations, $p = 10^{-4}$ for all except $p = 0.6 \times 10^{-3}$ for the difference of 0.3pt in F-measure), except the difference of 0.2pt in precision ($p = 0.104$). Note however that the methods and experiments presented in the present paper benefited from extra time invested after the official CLEF eHealth run submissions, so that a comparison with results obtained during the shared task time frame does not reflect differences in quality of the involved teams.

5 Discussion

5.1 Calibration

Calibration proved highly efficient in the present setting.

For instance, calibration of *boiu* output with only code-based classification (*boiu c* in Table 4) filters out 258 instances of ICD-10 code *C809,* Malignant neoplasms of ill-defined, secondary and unspecified sites which dictionary projection assigned to our test split, among which only 15 were true positives and 243 were false positives. The dictionary happens to have 509 entries for this code, among which the single word *cancer*. Because of the longest match strategy, this entry generally does not fire because longer entries including this word exist and will match instead. However, it acts as a default entry which may be used in inappropriate contexts.

Because we applied calibration to filter out some target codes, it blocks full sets of dictionary entries (for instance, the 509 entries for *C809*). A finer-grained method might try to filter out specific entries instead, and maybe still obtain a good increase in precision while limiting the associated loss in recall.

5.2 Union and intersection

Union and intersection are very simple combination methods. They played their expected roles in our experiments. Because we started from predicted results with precisions above 0.90, union was able to keep a high enough precision (up to 0.89 on the training set and 0.88 on the test set). The fact that it also led to a strongly increased recall shows that dictionary projection and our mono-label supervised classifier produced complementary results.

Given that we started from high-precision results, intersection was interesting to obtain very-high-precision classifiers. On the training set, the obtained precision ranged from 0.94 to 0.97, with associated recalls decreasing from 0.56 to 0.46. A study of the 3% resisting codes is left for future work. The highest-precision configuration, when applied to the test set, also reached a 0.97 precision with a 0.45 recall. This means that nearly one half of the test statements can be annotated automatically with an error rate of only 3%. This makes pre-annotation of death certificates with these methods a viable proposal to save human coding time.

5.3 Dictionary projection as a classifier feature

A very simple way to combine two classifiers is to use the output of one of them as a feature for the other. As suggested by an anonymous reviewer, we tested this scheme by using the ICD codes detected by dictionary projection (with the *boiu*, *best*, or *all* selection method) as an additional feature for the supervised classifier. We trained and tested the SVM supervised classifier with this additional feature based on the 2011 dictionary. This improved P, R and F by about 1pt on our test split (P=0.9154, R=0.6883, F=0.7858 with the *best* selection method). We then computed the intersection of the obtained classifier with the dictionary results (with and without calibration), as performed before to obtain the results in Table 5. This increased the best union F-measure (*all, all-c-t*) by up to 0.3pt at 0.8697 (with *all* selection method) as well as all

other union F-measures, but obtained a lower best intersection precision (–0.4pt at 0.9711, *boiu, boiu-c-t*). The influence of the selection method used in dictionary projection for feature creation was minor. This additional combination might increase again the F-measure on the test corpus, but was not tested in this paper.

5.4 Generalizability

The ICD-10 coding of death certificates is a process which is performed world-wide in a variety of languages. Efforts have been spent in various countries to develop dictionaries such as that of the CépiDc in France. An important feature of the methods we have presented here is that they are readily portable to other languages. The only language-dependent parts of our methods are diacritic removal (which generalizes to all Unicode languages to which the 'NFD' normalization applies), stemming (which is readily available for dozens of languages), some offline spelling correction (which generalizes to many alphabetic languages), and the use of character trigrams (which generalizes to alphabetic languages). No manual dictionary entry development or tuning was performed at all. Moreover, the supervised method already yields a high precision even without any dictionary at all, provided a sufficient number of training examples are available.

Therefore our methods and system should be applicable with no or little effort to a number of other languages.

6 Conclusion

We explored hybrid methods which combine simple dictionary projection and mono-label supervised classification. Our starting point was a dictionary projection method which obtained a higher recall and a supervised classification method which obtained a higher precision. Calibration strongly improved the precision of dictionary projection, making it higher than that of the supervised classifier. Union of calibrated dictionary projection results and supervised classification results improved the recall of both of them while keeping a high enough precision, leading to the highest F-measure on the training corpus. Intersection of calibrated dictionary projection results and supervised classification results obtained a record precision of 0.97 while producing codes for a little less than one half of the statements. This is a suitable configuration for automatic pre-annotation of death certificates which could save time to human coders. These experiments were performed on the training corpus: when applying the best development configurations to the test corpus, they led to three runs (F=0.8510–0.8586) which are all above the best published F-measure so far (F=0.848, significant at $p < 10^{-4}$) on this dataset. An important advantage of these methods is that they only relied on the data provided by the French coding center, CépiDc: if similar organizations in other countries have similar data, these methods should be readily applicable with little change to these new data.

In future work we plan to improve the individual methods and test more hybrid methods. Using more complete dictionaries is a way to improve the recall and maybe precision too of dictionary projection. Changing the supervised classification to perform multi-label classification is a direction to improve the recall of the supervised classifier. Calibrating the dictionary at the level of individual entries instead of target codes might also limit the loss of dictionary projection recall when increasing its precision.

Acknowledgments

We thank the CLEF eHealth challenge organizers for providing the data used in the present work and Jan Kors (ERASMUS team) for giving us access to their results for significance testing. This work was partially funded by the European Union's Horizon 2020 Marie Sklodowska Curie Innovative Training Networks—European Joint doctorate (ITN-EJD) Under grant agreement No:676207 (MiRoR). Finally, we thank the anonymous reviewers for their very relevant comments which helped improve the paper.

References

Alan R Aronson and François-Michel Lang. 2010. An overview of MetaMap: historical perspective and recent advances. *J Am Med Inform Assoc*, 17(3):229–36.

Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): Integrating biomedical terminology. *Nucleic Acids Research*, 32(Database issue):D267–270.

Mohammed Dermouche, V Looten, Rémy Flicoteaux, Sylvie Chevret, J Velcin, and Namik Taright. 2016. ECSTRA-INSERM @ CLEF eHealth2016-task 2: ICD10 code extraction from death certificates. In *CLEF 2016 Online Working Notes*. CEUR-WS.

Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, Ngan Nguyen, and Jun'ichi Tsujii. 2011. Overview of BioNLP shared task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 1–6, Portland, Oregon, USA, June. Association for Computational Linguistics.

Bevan Koopman, Sarvnaz Karimi, Anthony Nguyen, Rhydwyn McGuire, David Muscatello, Madonna Kemp, Donna Truran, Ming Zhang, and Sarah Thackway. 2015a. Automatic classification of diseases from free-text death certificates for real-time surveillance. *BMC Med Inform Decis Mak*, 15:53.

Bevan Koopman, Guido Zuccon, Anthony Nguyen, Anton Bergheim, and Narelle Grayson. 2015b. Automatic ICD-10 classification of cancers from free-text death certificates. *Int J Med Inform*, 84(11):956–965, November.

Claire Nédellec, Jin-Dong Kim, Sampo Pyysalo, Sophia Ananiadou, and Pierre Zweigenbaum. 2015. BioNLP Shared Task 2013: Part 1. *BMC Bioinformatics*, 16(Suppl 10), July.

Aurélie Névéol, Kevin Bretonnel Cohen, Cyril Grouin, Thierry Hamon, Thomas Lavergne, Liadh Kelly, Lorraine Goeuriot, Grégoire Rey, Aude Robert, Xavier Tannier, and Pierre Zweigenbaum. 2016. Clinical information extraction at the CLEF eHealth evaluation lab 2016. In *CLEF eHealth Evaluation Lab*.

John P. Pestian, Chris Brew, Pawel Matykiewicz, DJ Hovermale, Neil Johnson, K. Bretonnel Cohen, and Wlodzislaw Duch. 2007. A shared task involving multi-label classification of clinical free text. In *Biological, translational, and clinical language processing*, pages 97–104, Prague, Czech Republic, June. Association for Computational Linguistics.

Hanna Suominen, Sanna Salanterä, Wendy W. Sumitra Velupillai Chapman, Guergana Savova, Noemie Elhadad, Sameer Pradhan, Brett R. South, Danielle L. Mowery, Gareth J.F. Jones, Johannes Leveling, Liadh Kelly, Lorraine Goeuriot, David Martinez, and Guido Zuccon. 2013. Overview of the ShARe/CLEF eHealth evaluation lab 2013. In *Proceedings of CLEF 2013*, Lecture Notes in Computer Science, Berlin Heidelberg. Springer.

Özlem Uzuner, Yuan Luo, and Peter Szolovits. 2007. Evaluating the state-of-the-art in automatic de-identification. *J Am Med Inform Assoc*, 14:550–563.

Özlem Uzuner, Brett R. South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *J Am Med Inform Assoc*, 18(5):552–556, Sep-Oct. Epub 2011 Jun 16.

E Van Mulligen, Z Afzal, S A Akhondi, D Vo, and J A Kors. 2016. Erasmus MC at CLEF eHealth 2016: Concept recognition and coding in French texts. In *CLEF 2016 Online Working Notes*. CEUR-WS.

F. Wingert, David Rothwell, and Roger A Côté. 1989. Automated indexing into SNOMED and ICD. In Jean Raoul Scherrer, Roger A. Côté, and Salah H. Mandil, editors, *Computerised Natural Medical Language Processing for Knowledge Engineering*, pages 201–239. North-Holland, Amsterdam.

Xiaohua Zhou, Xiaodan Zhang, and Xiaohua Hu. 2006. MaxMatcher: Biological concept extraction using approximate dictionary lookup. In *Proceedings of the 9th Pacific Rim International Conference on Artificial Intelligence*, PRICAI'06, pages 1145–1149, Berlin, Heidelberg. Springer-Verlag.