Korektor – A System for Contextual Spell-checking and Diacritics Completion

Michal RICHTER[*] *Pavel STRAŇÁK*[*] *Alexandr ROSEN*[†]
[*]Charles University in Prague, Faculty of Mathematics and Physics
[†]Charles University in Prague, Faculty of Arts
{richter,stranak}@ufal.mff.cuni.cz, alexandr.rosen@ff.cuni.cz

Abstract

We present *Korektor* – a flexible and powerful purely statistical text correction tool for Czech that goes beyond a traditional spell checker. We use a combination of several language models and an error model to offer the best ordering of correction proposals and also to find errors that cannot be detected by simple spell checkers, namely spelling errors that happen to be homographs of existing word forms. Our system works also without any adaptation as a diacritics generator with the best reported results for Czech text. The design of *Korektor* contains no language-specific parts other than trained statistical models, which makes it highly suitable to be trained for other languages with available resources. The evaluation demonstrates that the system is a state-of-the-art tool for Czech, both as a spell checker and as a diacritics generator. We also show that these functions combine into a potential aid in the error annotation of a learner corpus of Czech.

Title and Abstract in Czech

Korektor – systém pro kontextovou opravu pravopisu a doplnění diakritiky

Představujeme *Korektor* – flexibilní statistický nástroj pro opravu českých textů, jehož schopnosti přesahují tradiční nástroje pro kontrolu pravopisu. *Korektor* využívá kombinace jazykových modelů a chybového modelu jak k tomu, aby setřídil pořadí nabízených náhrad pro neznámé slovo podle pravděpodobnosti výskytu na daném místě v textu, tak také, aby nalezl i překlepy, které se nahodile shodují s existujícím českým slovním tvarem. Prostou náhradou chybového modelu náš pracuje *Korektor* také jako systém pro doplnění diakritiky („oháčkování textu“) s nejvyšší publikovanou úspěšností. Systém neobsahuje žádné významné jazykově specifické komponenty s výjimkou natrénovaných statistických modelů. Je tedy možné jej snadno natrénovat i pro jiné jazyky. Ukážeme, jakých zlepšení náš systém dosahuje v porovnání se stávajícími českými korektory pravopisu i systémy pro doplnění diakritiky. Ukážeme také, že kombinace těchto schopností pomáhá při anotaci chyb v korpusu češtiny jako druhého jazyka.

Keywords: spellchecking, diacritics completion, language model, error model.

Czech Keywords: kontrola pravopisu, oprava pravopisu, doplnění diakritiky, jazykový model, chybový model.

---

1 Introduction

The idea of using context of a misspelled word to improve the performance of a spell checker is not new (Mays et al., 1991), moreover, recent years have seen the advance of context-aware spell checkers such as *Google Suggest*, offering reasonable corrections of search queries. Errors detected by such advanced spell checkers have a natural overlap with those of rule-based grammar checkers – grammatical errors are also manifested as unlikely n-grams.

Methods used in such spell checkers usually employ the *noisy-channel* or *winnow-based* approach (Golding and Roth, 1999). The system described here also belongs to the *noisy-channel* class. It makes extensive use of language models based on several morphological factors, exploiting the morphological richness of the target language.

The purpose of this work was to implement a flexible system, capable of performing diverse tasks such as spelling correction, diacritics completion and abbreviated text expansion by simple module replacement. Rather than presenting a scientific prototype we aimed at a practical system providing a better spell-checking for Czech than systems currently available.

Section 2 introduces the statistical models used here and describes their application to the tasks of spell-checking and diacritics completion. In Section 3 results of performance evaluation are presented. Section 4 discusses the system's performance, while Section 5 provides conclusions and outlines our plans for the future.

2 Statistical Model

The task of context-sensitive spelling correction and diacritics completion can be seen as a problem of sequence decoding, which is often formulated in terms of the noisy-channel model. A transmitter sends a sequence of symbols to a receiver. During the transfer, though, certain symbols of the transmitted sequence are garbled due to the deficiencies of the transmission channel. The receiver's goal is to reconstruct the original sequence using the knowledge of the *source* (i.e. how a Czech sentence looks like) and the transmission channel properties.

2.1 Source Modeling

Several feature functions[1] were used to model the source:

- Word forms feature F_f – based on language model probability $P(f_i|f_{i-2}, f_{i-1})$, where f_i denotes the next word form and f_{i-2} and f_{i-1} are the previous word forms.
- Morphological lemma feature F_l – based on language model probability $P(l_i|l_{i-2}, l_{i-1})$ and emission model probability $P(f_i|l_i)$, where l_i stands for the next lemma, l_{i-1}, l_{i-2} are the previous lemmas and f_i is the next word form.
- morphological tag feature F_t – which is, analogically to the morphological lemma feature, based on $P(t_i|t_{i-2}, t_{i-1})$ and $P(f_i|t_i)$.

The source feature functions are task independent. Their role is to approximate grammaticality of the output sentence. The probability measures were estimated on the basis of n-gram

[1] For the convenience of the reader, the feature functions are based on trigram statistic in the descriptions. However, higher order n-grams are supported as well.

counts collected from a training corpus, using interpolated Kneser-Ney (Kneser and Ney, 1995) smoothing.

A large text corpus was needed in order to produce well-estimated language models and word emission models. This need was met by the *Czech Web Documents Collection* (henceforth *WebColl*) (Marek et al., 2007), a 111 million words resource consisting of 223,000 articles, downloaded from news servers and on-line archives of Czech newspapers, lemmatized and tagged with detailed morphological tags as described in the paper. N-gram counts for each morphological factor and counts of `form-lemma` and `form-tag` combinations were collected. For the word forms and lemmas, n-grams up to order 3 were collected. For morphological tags, 4-grams were collected as well.

2.2 Channel Modeling

A single channel feature F_{ch} estimates the probability $P(f|f^{'})$ of word form $f^{'}$ being transferred as word form f. The channel feature links the input and the output words. The score is assigned according to the similarity of the output words to the input words according to the task specific similarity measure – for the spelling correction problem, it takes into account the probabilities of specific typing errors. Transmission channel for the diacritics completion is constructed in such a way that it assigns a uniform cost to all variants of an output word with diacritics and the infinite cost to all other words.

2.3 Log-linear Model and Viterbi algorithm

A log-linear model (Jurafsky and Martin, 2008) was used to combine all feature functions into a single statistical model. The search space of the model is enormous – $|V|^{|S|}$, where $|V|$ is the vocabulary size and $|S|$ is the sentence length. However, since all the features use only limited history, we could use Viterbi algorithm (Viterbi, 1967) to find the optimal hypothesis.

2.4 Error Model For Spelling Correction

The error model used in this work is based on the model of (Church and Gale, 1991). They consider only candidate words obtained by a single edit operation – insertion, deletion, substitution, or swap. This model is a good fit for Czech language. Since Czech has mostly phonetic spelling, the errors tend to be local, limited to one of these operations. Edit operations have their distinct probabilities, i.e. the probability of the letter substitution $s \rightarrow d$ may differ from the probability of $e \rightarrow a$. Letter insertion and deletion probabilities are also context-conditioned.

These probabilities were estimated from the large text corpus. They considered each word that did not appear in the dictionary and was not farther than one edit operation from a word included in the dictionary as a spelling error, and built their error corpus out of such words. First, they set the probabilities of all edit operations uniformly. Later on, they iteratively spell-checked their error corpus, found the best correction for each word and updated the edit probabilities according to the proposed $error \rightarrow suggestion$ pairs.

This method of finding spelling errors was tested on the *WebColl* corpus (see Section 2.1), but turned out to be useless. The reason was that the vast majority of words identified as spelling errors were correct words or colloquial word forms.

The modified version builds an error corpus out of words recognized by the spell checker as spelling errors, however there must be a significant evidence that the proposed correction is

Error Type	Cost	Error Type	Cost
Substitution – horizontally adjacent letters	2.290	Substitution – diacritic redundancy	2.250
Substitution – vertically adjacent letters	2.661	Substitution – other cases	4.285
Substitution – $z \rightarrow s$	2.747	Insertion – horizontally adjacent letter	2.290
Substitution – $s \rightarrow z$	1.854	Insertion – vertically adjacent letter	2.661
Substitution – $y \rightarrow i$	3.167	Insertion – same letter as previous	1.227
Substitution – $i \rightarrow y$	2.679	Insertion – other cases	2.975
Substitution – non-adjacent vocals	3.706	Deletion	4.140
Substitution – diacritic omission	2.235	Swap letters	3.278

Table 1: Spelling Error Types together with their costs ($-log$ of their probabilities)

right, otherwise the spelling error is not added to the error corpus. More specifically, both bigrams (w_{i-1}, s) and (w_{i+1}, s), where w_{i-1} is the predecessor of a misspelled word e, w_{i+1} is the successive word and s is the correction suggestion, must be present in the language model, otherwise the error-correction pair $e \rightarrow s$ is not included in the error corpus. Recall of this method is rather small, but the precision is quite satisfactory and most of the recognized error-correction pairs were correct. This method identified 12,761 words out of 111,000,000 words in *WebColl* as spelling errors. A classification of these errors is shown in Table 1. The granularity of spelling error types being distinguished is much smaller than in (Church and Gale, 1991).

2.5 Letter Language Model For Diacritics Completion

It may happen that for a given word of the input sentence, no candidate word is found. An example of such a word is *nemeckofrancouzsky* 'German-French', which remains untouched without any added diacritics. However, an error, here a missing hyphen, is very likely. The present example should receive diacritics as in *německofrancouzský* (adjective) or *německofrancouzsky* (adverb).

In order to cut down the number of errors made on unknown words, a custom implementation of the Viterbi decoder was provided. The states on the underlying HMM are tuplets of letters and the transition probabilities are given by a letter n-gram language model (it estimates the probability of next letter on the basis of previous letters). The aim of this Viterbi decoder is to find the most probable letter sequence given the input letter sequence. The only substitutions allowed are the substitutions that add diacritics. Using this approach, diacritics can be added correctly even to the unknown words.

Given that the vocabulary of letter n-gram language model is extremely small (the size of the alphabet), it is possible to train letter LMs of a very high order. In this work, letter LMs of the order up to 7 were trained. The letter LMs were trained on the training part of *WebColl*.

During the evaluation the contribution of using letter LMs was examined. Table 4 shows a significant accuracy improvement when this feature is used.

3 Evaluation

3.1 Diacritics Completion Results

Diacritics completion was evaluated on three different data sets: the development part of the *WebColl* corpus and the Czech translations of two books: by Martin Gilbert's *A History of the Twentieth Century* (non-fiction) and Lion Feuchtwanger's *Foxes in the Vineyard* (fiction). All the diacritics in the testing data were simply removed. Then the system generated it back and the

α_l	non-fiction	fiction	*WebColl*
0.1	97.45%	96.82%	98.16%
0.3	97.49%	96.86%	98.21%
0.5	97.51%	96.85%	98.20%
0.7	97.45%	96.77%	98.09%
0.9	97.18%	96.48%	97.79%

Table 2: Results of *form – lemma* experiments. Only F_l and F_f are used for source modeling and $\alpha_f = (1 - \alpha_l)$.

α_t	non-fiction	fiction	*WebColl*
0.1	97.66%	97.20%	98.35%
0.3	97.83%	97.60%	98.53%
0.5	97.88%	97.74%	98.57%
0.7	97.85%	97.71%	98.52%
0.9	97.62%	97.53%	98.26%

Table 3: Results of *form – tag* experiments. Only F_t and F_f are used for source modeling and $\alpha_f = (1 - \alpha_t)$.

data set	α_f	α_l	α_t	accuracy – no Letter LM	accuracy – with Letter LM
non-fiction	0.31	0.28	0.41	97.9%	98.3%
fiction	0.31	0.14	0.55	97.7%	97.9%
WebColl	0.34	0.33	0.33	98.6%	99.1%

Table 4: The best accuracy values achieved on each testing set.

results wece compared to the originals.

The main parameters are the weights α_f, α_l and α_t of features F_f, F_l and F_t.

First, the contributions F_l and F_t were examined separately. In these experiments α_f was ranging from 0 to 1 and the weight $(1 - \alpha_f)$ was given either F_l or F_t, all the language models used were trigrams. The results of such experiments are shown in Tables 2, 3. It is clear from the plots that both features F_l, F_t improve the system performance. However the contribution of F_t is more significant. Surprisingly, it seems to be better to give all the weight to F_t than to give all the weight to F_f.

The performance boost achieved by using F_t is most visible on a comparison of results achieved on history domain and fiction domain data. For baseline setup ($\alpha_f = 1, \alpha_t = 0$), the accuracy is 97.39% on non-fiction data and 96.74% on fiction data, which means that the error rate is 25% bigger on fiction data. Nevertheless, by increasing the weight of F_t the difference in performance was becoming less significant and for the best parameter settings ($\alpha_f = 0.4, \alpha_t = 0.6$), the error rate on fiction data was only 7% bigger (97.72% accuracy on fiction data and 97.89 on non-fiction data).

Next, the estimation of the best parameter setting for each data set was done using a simple hill-climbing algorithm (see (Russell and Norvig, 2003) for details). As the starting point, all the weights were set equally. The resulting parameters and the accuracy values are shown in the Table 4. Experiments with the letter LM feature turned on were made also for the particular settings. It can be seen that the use of letter LM for the completion of unknown words improves the results significantly.

Results of the diacritics completion provided by *Korektor* were compared with those of *CZACCENT*[2], the diacritics completion tool developed by the NLP Center of Masaryk University in Brno, using the non-fiction data set. The accuracy achieved by *CZACCENT* was 95.85%, while the accuracy achieved by *Korektor* reached 98.3%. The error rate of *Korektor* is thus almost 2.5 times smaller.

[2]http://nlp.fi.muni.cz/cz_accent/index.php

3.2 Spell-checking Results

The quality of spell checkers is usually measured by the spelling correction error rate (i.e., the probability that the first given suggestion is correct, or that the correct suggestion is included in the list of first three suggestions, etc.). If a context-sensitive spell checker is considered and the ability of recognizing the real-word errors is to be tested, *F-measure* based on *precision* and *recall* can be used. It is a good indicator of a quality of a classifier.

During the evaluation of spelling correction, the optimal parameter settings (weights of distinct feature functions), estimated for the diacritic completion task, were used on the assumption that the features F_f, F_l and F_t are task independent and that their weighting obtained for one task will perform well for other tasks as well. The reason why we made no separate parameter tuning was that the size of available annotated spelling error data was too small. The weights were set according to the optimal setting for diacritics completion on non-fiction, i.e. $\alpha_f = 0.31$, $\alpha_l = 0.28$ and $\alpha_t = 0.41$. Channel feature F_{ch} was set to the weight $\alpha_{ch} = 1.0$, which assigns the same importance to both the source and the channel models.

For the evaluation of spell-checking, three different data sets were used: 1. *Chyby* – an error corpus (Pala et al., 2003); 2. *Audio* – transcription of an audio book; 3. *WebColl* test set – semi-automatically recognized spelling errors in the part of *WebColl* not used during the training. 4. *CzeSL* – a corpus of short essays written by learners of Czech as a foreign language

The error corpus *Chyby* (Pala et al., 2003) is a collection of essays written by students of Brno University of Technology, annotated for errors including spelling, morphological, syntactic and stylistic errors. Spell checking was tested on spelling and morphological errors since these types of errors are potentially recognizable by the system. There were 744 such errors, 321 of them were real-word errors. The high ratio of real-word errors show that most of the student works were already spell-checked.

The *Audio* test set, including the total of 1371 words, has 218 spelling errors, 12 of them real-word errors. The data set was built by transcribing an audio version of Jaroslav Hašek's novel *Osudy dobrého vojáka Švejka* 'The Good Soldier Švejk'.[3] The transcribed text was not post-corrected and the spelling error rate in the resulting text is relatively high.

The *WebColl* testing set was extracted from the part of *WebColl* not used during the system training. Spelling errors were collected semi-automatically using *Korektor*. Words identified by the spell-checker[4] as spelling errors were examined manually and the words that were flagged as spelling errors by mistake were filtered out. The result was a set of sentences containing spelling errors authorized by a human. The golden standard data were created manually in the next step. This approach made the collection of errors in the *WebColl* testing data feasible, but all real-word errors were missed (they were ignored, because they were not flagged as spelling errors by the spell checker in the first step). Thus, only the evaluation of suggestion accuracy could be done for this data.

The results of spelling correction accuracy evaluation for *Chyby*, *Audio* and *WebColl* are shown in Table 5 and the results of real-word error detection evaluation are shown in Table 6. For

[3] The audio extracts can be downloaded for free from the website of the Czech Radio: `http://www.rozhlas.cz/ctenarskydenik`.

[4] The spell checker made look-up for the out–of–vocabulary words easier. The correction suggestions provided by the spell checker were not taken into consideration during the creation of the golden standard data, so the fact that the spell checker to be tested participated in the creation of the testing set does not invalidate the testing set.

Number of suggestions	*WebColl*	*Chyby* 1	*Chyby* 2	*Audio* (*Korektor*)	*Audio* (*MS Word*)
1	91.4%	73.5%	82.3%	91.6%	71.2%
2	95.1%	80.1%	80.9%	97.2%	-
5	96.3%	80.9%	90.5%	98.6%	-

Table 5: Spelling correction rates achieved on the different data sets. For the *Chyby* corpus, two measurements were taken. In *Chyby 1*, all spelling errors are considered. For *Chyby 2*, only those spelling errors for which an appropriate correct version is in the lexicon are taken into account.

	Chyby	*Audio* (*Korektor*)	*Audio* (*MS Word*)
Precision	0.41	1.0	0.5
Recall	0.24	0.77	0.08
F-measure	0.31	0.87	0.14

Table 6: Real-word error correction statistics for *Audio* data set and *Chyby* corpus.

the *Audio* data set, comparison with the *Microsoft Word 2007* spell checker with grammar checking features turned on was made. For *MS Word* spell checker only the accuracy on the first suggestion was considered since there is no API that would allow to do the evaluation automatically. We are not certain how exactly the MS system works, since as far as we know no details have been published. However, we think that it is a conventional spell checker without any statistical model (for Czech) and a rule based grammar checker. Since the components seem to be separate, the grammar checker assumes the text has already been spell checked. This assumption, combined with what looks like a minimum edit distance algorithm to pick the first suggestion of the spelling module provides a disadvantage for MS Word system in the fully automated setting.

The results suggest that *Korektor* has a much higher accuracy on a single suggestion and ability to detect real-word spelling errors. The cases when the *MS Word* spell checker marked a grammar error were all because of capitalization problems, which suggests that there is no statistical real-word error detection in the Czech version of *MS Word* [5] Significantly lower spelling correction rate on the *Chyby* corpus can be caused by the fact that the properties of the texts in this corpus (technical topics) differ significantly from the training data properties (newspapers).

Finally, *Korektor*'s performance was tested on a sample from *CzeSL*, a learner corpus consisting of texts produced by learners of Czech as a second or foreign language. A part of the corpus is manually annotated in two stages with correct versions of deviant forms and relevant error codes. The annotators are instructed to correct both non-words (stage/Tier 1) and real-word errors (stage/Tier 2) to arrive at a grammatically correct sentence.[6]

In a pilot study of 67 short, doubly-annotated essays, *Korektor* was used to see whether automatic correction of learner texts is viable as a way to assist the annotator or even as a fully automatic annotation procedure.

Among the total 9,372 tokens, 918 (10%) were not recognized by a tagger (Spoustová et al., 2007) we used to find incorrect word forms. Even more forms were judged as faulty by the

[5]However, the *MS Word* spell checker for Czech is equipped with other capabilities that *Korektor* does not possess, such as punctuation checking.

[6]See (Hana et al., 2010).

annotators: 1,189 (13%) were corrected in the same way by both annotators at Tier 1 (T1) and 1,519 (16%) at Tier 2 (T2). Results of *Korektor* were compared with those of the tagger and with forms at T1 and T2, provided both annotators were in agreement. In the case of the tagger *Korektor* was deemed to be successful if it agrees with the tagger in the correct/incorrect status of the form. The results in terms of F-measure show 0.86 in comparison with the tagger, 0.72 in comparison with T1 and 0.53 in comparison with T2. The results support the idea to integrate *Korektor* into the learner corpus annotation workflow, either as suggestions to the annotator or as a solution to obtain fully automatic large-scale annotation at the cost of a higher error rate. In fact, the entire *CzeSL* corpus (2 mil. words, including unannotated parts) has been processed by *Korektor* to help querying the corpus.

4 Discussion

The results for spelling correction accuracy are not as good as those reported in (Brill and Moore, 2000) – around 95% on the first suggestion. However, those results were achieved for English and are not directly comparable. Czech with its rich morphology may be more challenging. For the *Chyby* corpus, significantly lower performance (73% on the first suggestion) may be caused by the heavy usage of technical terminology, such as names of software products, including their inflected forms. On the other hand, the fact that *Korektor* clearly outperformed the spell checker integrated in *Microsoft Word 2007* indicates the qualities of the system.

5 Conclusion and Future Work

We have designed and implemented a context-sensitive method of spell-checking and diacritics completion. The result is a spell checker that is freely available and ready for use.

Our primary concern was a robust, purely statistical, language-independent design. As a result, the system can be re-trained for any language. The only limitation is the availability of an annotated error corpus to train the error model, the availability of a general corpus to train the language model, and (depending on the language) a lemmatizer / POS tagger.

As for the spell-checking task, we focussed on the ability of the system to recognize real-word spelling errors and also to suggest the most likely corrections of spelling errors. In the spell-checking evaluation, *Korektor* achieved much better performance than the *MS Word 2007* spell checker.

Diacritics completion module was implemented on top of the spell checker. The accuracy of diacritics completion was about 98% with training and test data coming from different domains. Such performance is acceptable for many tasks, the best reported for Czech so far, and among the best reported for any language.

Korektor was also applied to texts produced by non-native speakers of Czech to provide anotation of a learner corpus. The result will soon be available for on-line searching via a concordancer.

In the future, we want to train *Korektor* for other languages by creating language and error models for the individual languages. In that setting a possible improvement could be achieved by utilization of more fine-grained error models as proposed by (Brill and Moore, 2000). In standard Czech it has a limited value as explained in Section 2.4, but the experiments on a learner corpus show that even in Czech it could still be useful for non-native speakers. For languages with less straightforward orthography, such as English, it would be even more valuable.

References

Brill, E. and Moore, R. C. (2000). An improved error model for noisy channel spelling correction. In *ACL '00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 286–293, Morristown, NJ, USA. Association for Computational Linguistics.

Church, K. and Gale, W. (1991). Probability scoring for spelling correction. *Statistics and Computing*, 1(7):93–103.

Golding, A. R. and Roth, D. (1999). A winnow-based approach to context-sensitive spelling correction. *Machine Learning*, 34:107–130. 10.1023/A:1007545901558.

Hana, J., Rosen, A., Škodová, S., and Štindlová, B. (2010). Error-tagged learner corpus of Czech. In *Proceedings of the Fourth Linguistic Annotation Workshop*, Uppsala, Sweden. Association for Computational Linguistics.

Jurafsky, D. and Martin, J. H. (2008). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, second edition.

Kneser, R. and Ney, H. (1995). Improved backing-off for M-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184 vol.1.

Marek, M., Pecina, P., and Spousta, M. (2007). Web page cleaning with conditional random fields. In Fairon, C., Naets, H., Kilgarriff, A., and de Schryver, G.-M., editors, *Proceedings of the 3rd Web As a Corpus Workshop, Incorporating CLEANEVAL*, pages 155–162, Louvain-la-Neuve, Belgium. UCL Pressess Universitaires de Louvain.

Mays, E., Damerau, F. J., and Mercer, R. L. (1991). Context based spelling correction. *Information Processing & Management*, 27(5):517 – 522.

Pala, K., Rychlý, P., and Smrž, P. (2003). Text corpus with errors. In *TEXT, SPEECH AND DIALOGUE*, volume 2807/2003 of *Lecture Notes in Computer Science*, pages 90–97. Springer.

Russell, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition.

Spoustová, D., Hajič, J., Votrubec, J., Krbec, P., and Květoň, P. (2007). The best of two worlds: Cooperation of statistical and rule-based taggers for Czech. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing 2007*, pages 67–74, Praha, Czechia. Association for Computational Linguistics.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269.