ПОИСК И РАНЖИРОВАНИЕ ИЛЛЮСТРИРУЮЩИХ ПРИМЕРОВ ДЛЯ ПЕРЕВОДНОГО СЛОВАРЯ

Протопопова Е. (rhubarb@yandex-team.ru),
Антонова А. (antonova@yandex-team.ru),
Мисюрев А. (misyurev@yandex-team.ru)

Яндекс, Москва, Россия

Ключевые слова: автоматическое создание словарей, параллельный конкорданс, векторные модели

ACQUIRING RELEVANT CONTEXT EXAMPLES FOR A TRANSLATION DICTIONARY

Protopopova E. (rhubarb@yandex-team.ru),
Antonova A. (antonova@yandex-team.ru),
Misyurev A. (misyurev@yandex-team.ru)

Yandex, Moscow, Russia

This paper addresses the problem of automatic acquisition of parallel context examples for a translation dictionary. We extract them automatically from a parallel corpus, relying on word alignments and parse trees. The ranking of the extracted examples is an essential problem, since we need to select the most distinctive and informative contexts. We propose a machine learning approach as an alternative to simple ranking criteria, such as frequency, or mutual information. We perform the analysis of common sources of inadequate context examples and design a set of features, which can possibly distinguish the bad examples from the good ones. We also experiment with vector models (word2vec) in order to get features that are sensitive to semantics. The evaluation result show that the best of our ranking methods yields 31% improvement in accuracy compared to the ranking by frequency, and 20% improvement over the ranking by mutual information. Using vector models also improves the classification performance.

Keywords: bilingual dictionary extraction, bilingual concordance, vector models

1. Introduction

The paper is concerned with a problem of automatically acquiring the illustrative translation examples for English-Russian machine dictionary. Such examples can enrich the dictionary entry, illustrate semantic and syntactic selectional preferences, and help the user to differentiate between the meanings of multiple translation variants. Many well-known translation dictionaries include examples, which had been prepared by professional lexicographers.

Recently, the growing amount of parallel documents in the Internet and the current progress in language processing algorithms makes it possible to retrieve the context examples automatically from large-scale parallel corpora. Figure 1 shows how automatically extracted context examples are used to illustrate different meanings of the words *'French'* and *'пример' ('example')* in an online dictionary.

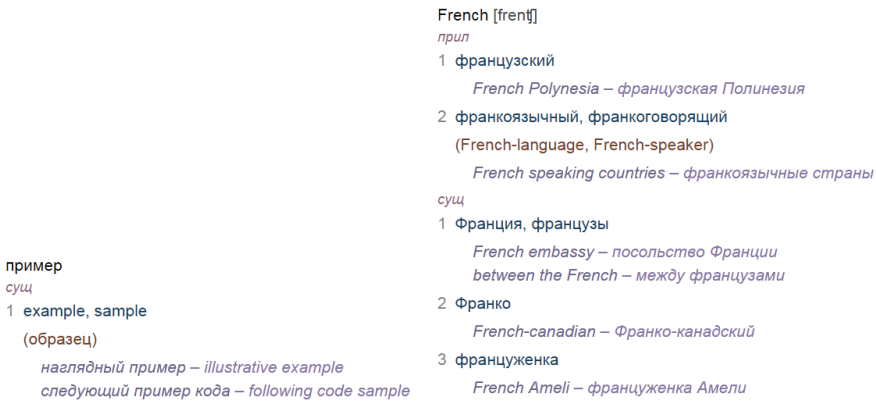

Fig. 1. Illustrative examples in a bilingual machine dictionary

The dictionary format imposes the following requirements on the context examples:

- Only one or several best examples are shown per one translation.
- Examples should be short well-formed grammatical phrases.
- Examples should represent a characteristic use of a given word or expression.

Examples are extracted from parallel sentences where a given translation pair is found with the help of word alignment (acquired by GIZA++ [9]) and a phrase extraction algorithm [6]. The sentences are processed by a dependency parser [1] so that we can search for words in different forms. Only phrases constituting a connected subgraph of a sentence parse tree are extracted and thus most of the ungrammatical phrases are discarded. This step is discussed in [2]. Parallel corpus is compiled from web-archives of a commercial search engine.

The essential problem is the ranking, since we need to eliminate all kinds of noisy contexts and select the most distinctive and informative ones. There exist simple ranking criteria, such as frequency, or mutual information, but they do not always work well. For example, when a phrase frequency is taken into account, then frequent but

useless examples are often ranked best (*then <go> → затем <перейти>*). If we use a metric like mutual information, too specific examples can be scored better (*unpregnant <woman> → небеременная <женщина>*).

In this paper we propose a machine learning approach to the ranking problem. We analyse typical mistakes and design a set of features, which can possibly distinguish the bad examples from the good ones. We also use features from vector models (word2vec tool [8]) in order to predict syntactic and semantic relatedness between words.

We report on the experiments with two-words examples (bigrams). Different classifiers are trained on a manually annotated sample of automatically extracted examples. The classifiers' scores are used for elimination of noisy examples and for ranking the remaining ones. In some experiments the remaining examples are ranked according to a simple measure such as frequency. We also try to estimate prediction confidence using a combination of classifiers in order to find the most relevant examples.

The results are evaluated as follows. We compute precision, recall and accuracy of the classification using an annotated test set. We also perform a comparative evaluation of the accuracy of one-best examples found by different methods. The best of our ranking methods yields 31% improvement in accuracy compared to the ranking by frequency, and 20% improvement over the ranking by mutual information.

The advantages of automatic approach to the task of creating context examples are the following:

- The automatic approach enables us to find up-to-date and frequently used phrases.
- The procedure can be repeated on bigger or different corpora in order to cover more meanings and words.
- Our statistical approach can be applied to any language pair with available corpora and a syntactic parser.

The paper is organized as follows. In Section 2 we briefly outline the related work. Section 3 describes the principles and the results of the examples annotation. Then we discuss the classification task in Section 4. Section 5 is devoted to classification experiments and system evaluation.

2. Related work

The papers concerning bilingual lexicon acquisition pay little attention to the problem mentioned in this paper, but task in general corresponds to that of building a bilingual concordance, i.e. finding all the examples of the word usage in text with their respective translations. Such systems are intended for translators and language learners. In some papers ([5], [7]) the issue is reduced to finding all sentences with a given source word and the presented systems do not take into account target expression and do not extract smaller phrases.

Ranking is not of great importance when building a bilingual concordance. Some of the systems such as the one discussed in [4] provide user with frequency information about collocations. In [10] the system ranks sentences and their translations according to frequency statistics, while the authors of [3] use Dice coefficient to show more relevant translations first.

3. Examples annotation

The classification task requires annotated data for learning, so first of all the data for annotation should be prepared. In this section we describe our experimental set as well as the principles of annotation.

3.1. Selecting translation pairs and examples

In order to make training and test sets more representative we try to select translation pairs and the respective examples so that their frequency distribution reflects the real word frequency distribution in parallel corpora. It is also important to illustrate source words which are more frequently queried in machine dictionary. We have noticed [2] that the amount of queries for source words highly correlates with source word frequencies, so we can rely on corpus statistics when selecting pairs for annotation. Finally, we create a random sample of English words excluding the most frequent hundred.

Each source word has one or several translations (target expressions) in our dictionary. For each pair 'source word—target expression' we extract all possible context examples from a web-based parallel corpus. However, random sampling from all examples would be quite unreliable because it would not ensure the balance between relevant and irrelevant examples. Thus, for each translation pair we select several best examples according to source and target frequencies as follows:

$$F = log(f_3) - log(f_1) - log(f_2)$$

where f_1 and f_2 are frequencies of words which do not form a given translation pair and f_3 is the whole example frequency.

3.2. Annotation principles

Table 1. Annotation principles

score	both sides annotation	one-side annotation
1	both parts are meaningless and grammatically incorrect; the parts are not translation equivalents	a phrase is meaningless and grammatically incorrect
2	one of the parts can be scored with 1 in one-side annotation or one or both parts are grammatically incorrect	a phrase is grammatically incorrect; a phrase is not a translation equivalent
3	both parts are grammatically correct but do not reflect any peculiarities of the translation pair	a phrase is grammatically correct but does not reflect any peculiarity of a word/expression
4	both parts are correct and partially illustrate peculiarities of a given pair	a phrase is correct and partially illustrates peculiarities of a given word/expression
5	relevant example	relevant example

The machine dictionary is created automatically and contains some noise. These noisy translation pairs and the respective examples are removed from the annotation set. Then we perform two kinds of annotation: assessing the whole example and assessing its source and target phrases separately. In each case we assign a score which ranges from 1 (very bad) to 5 (excellent). Table 1 specifies the requirements for all scores. The examples scored with 3 are then removed from the training set, as they are neither negative, nor positive.

3.3. Annotation results

After annotating 700 bigram examples we remove phrases extracted for incorrect translation equivalents. The number of examples for each score is shown on Figure 2. The number of erroneous Russian examples is somewhat higher because of the higher number of grammatical mistakes (see Section 3.4). As a whole, more positive examples were extracted due to filtering by frequency.

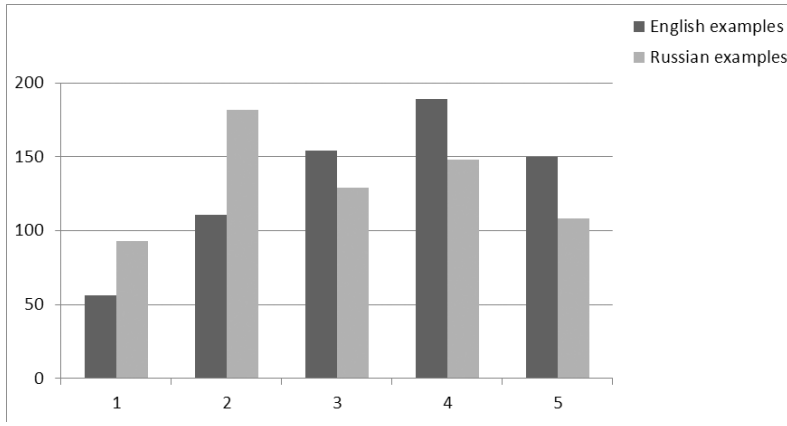

Fig. 2. The distribution of scores

3.4. Error analysis

The following errors are observed in automatically extracted examples (source and target expressions are marked with angle brackets, errors are marked with an asterisk):

1. Inadequacy in surface form
   (a) Ungrammatical phrases
   **<preparation> enamel → <составление> эмали*
   *<appreciate> acrobatics → *<оценить> акробатика*
   (b) Incomplete phrases
   county <detention> → деревенский <исправительный>

(c) Phrases not in dictionary form
**<created> tsunamis → *<породило> цунами*
*monstrously <big> → *чудовищно <огромная>*
*header files → *заголовочных файлов*

(d) Phrases containing a foreign word
*<improve> resiliency → *<улучшать> resiliency*
*unformatted <capacity> → *unformatted <емкость>*
**<beginning> shvatyvanija → начало> схватывания*

(e) Phrases containing a misspelled word
*caribbean <community> → *караибское <содружество>*
*burgundy <sole> → *бардовая <подошва>*

2. Inadequacy in meaning

(a) Uninformative phrases
**его <любовь> → *his <fondness>*
**очень <глупый> → *really <stupid>*
**nonpregnant <woman >→ *небеременная <женщина>*

(b) Phrases with unrelated words
**pickled <loveliness> → *маринованная <красота>*
**<saving> neurotic → *<спасение> невротиков*
**синхроничная <жизнь> → *synchronistic <life>*

(c) Hardly understandable phrases with specific meaning
**sagittal <reconstruction> → *сагиттальная <реконструкция>*
**threshold <panel> → *пороговое <табло>*

(d) Machine translation
**<soya> squirrels → <соевый> белок*
**<character> stitches → <символьные> строчки*
**harvest <control> → жмешь <контрол>*
**Berners-<whether> → Бернерс-<ли>*
*hi <camcorder> → *привет <видеокамеры>*

(e) Offensive contexts for neutral words
naked <girl> → голая <девушка>
<Japanese> militarists → <японские> милитаристы
Hitlerite <Germany> → гитлеровская <Германия>
<become> a Shaheed → <стать> шахидом

(f) Phrases which are not translations of each other
<saving> rolling → <спасение> утопающих

The first group of errors can be explained by the fact that almost no limitations are placed on extracted parse subtrees. This problem may be overcome by means of special rules which filter out some ungrammatical translations. Parallel machine translated sentences and misspelled words are frequent on websites and can be drawn when gathering parallel corpus in the internet. In some cases the sentences in the target text contain only partial translation of the source sentences, and phrases from them are also extracted as context examples.

4. Classification

4.1. Feature sets

We propose several groups of features which can distinguish the irrelevant examples from the informative ones.

Language model scores (LM)

Language models are concerned with example fluency as well as with filtering out grammatically incorrect expressions. We use English and Russian trigram language models compiled on big monolingual corpora containing Web documents. We also build part-of-speech trigram models using the sequences of morphological tags acquired by a statistical parser. We compute the following values:

- example perplexity according to unigram LM (2 features);
- example perplexity according to trigram (bigram in case of bigram examples) LM (2 features);
- the scores mentioned above using part-of-speech LM (2 features).

Relative Frequency (RelF)

We use the example frequency as described in Section 3:

$$RelF = log(f_3) - log(f_1) - log(f_2)$$

where f_1 and f_2 are frequencies of words which do not form a given translation pair and f_3 is an example frequency.

Mutual information (MI)

The average mutual information score for bigrams is computed for both sides of example treating two words as bigram if there is a syntactic link between them:

$$MI = log \frac{f(w_1, w_2)}{f(w_1) f(w_2)}$$

where $f(w)$ is the relative frequency of word w in a corpus and $f(w_1, w_2)$ is the relative frequency of the pair (w_1, w_2) connected with an arc in a parse tree. The relative frequencies are extracted from monolingual corpora annotated with the help of a statistical parser. Thus we can find more idiomatic examples with less frequent words.

Semantic similarity (Sim)

Word vectors computed by word2vec tool [8] on a large monolingual corpus have proved to be very efficient in capturing different linguistic regularities. We try to exploit them to find out more typical and specific word usages. Using word2vec tool we represent each word by a 200-dimensional word vector. Then we compute the cosine similarity measure in each one-side example. In case of three or more words we suggest calculating average similarity between all vectors as well as similarity between a given word and all other words in an example. This results in two features, one for each example side.

Vector models (WV)

As mentioned above, each word can be represented as a semantic vector, which can be used for training as is. We concatenate all vectors for words in a one-side example and also introduce binary features to indicate a key word for an example. Thus a feature vector for a two-word example (u,v) where the key is the second word looks like $(u_1, ..., u_{200}, v_1, ..., v_{200}, 0, 1)$ which means that 402 features are used. Concatenation requires that examples of different length are trained separately.

4.2. Classifiers

Simple binary classification

The examples annotation is quite detailed and quite difficult to predict automatically, so first of all we build a binary classifier to distinguish between informative and irrelevant or erroneous examples. We use a Random forest classifier as well as a feed-forward neural network with a single hidden layer.

Estimating prediction confidence

The multilabel annotation is useful when we try to find the examples which are undoubtedly relevant. For this purpose we combine four binary random forest classifiers for each score excluding examples with the closest score from the training set, for instance, when treating the 4th class as positive examples, we remove all examples of the 5th class and leave 1st and 2nd classes as negative examples. When predicting scores on test set, we use all classifiers and choose that with the highest predicted value and estimate confidence c as

$$c = \left| max(f_1, f_2) - max(f_4, f_5) \right|$$

where f_i is a predicted value of i-th classifier.

5. Test data and experimental setup

5.1. Assessing classifiers

Table 2. Classifiers performance. P_0 is the precision on negative examples and A is the classification accuracy

	$P_0(en)$	$A(en)$	$P_0(ru)$	$A(ru)$
RF_1	0.71	**0.74**	0.62	**0.64**
RF_2	**0.83**	0.65	**0.63**	0.62
NN	0.67	0.65	0.56	0.61

One-side prediction

	P_0	A
RF_3	0.690	0.70
RF_4	0.685	0.71

Both sides prediction

For each of the 52 random English words sampled according to the frequency distribution paired with all possible Russian translations from an online machine dictionary [2] we extract 3 best examples according to both sides frequencies and annotate the resulting examples removing those for incorrect translations. We split the resulting set into training (416 examples) and test (206 examples) parts. Firstly, we perform classification for source and target side separately using the following combinations:

- RF_1—random forest classifier using *WV* features;
- RF_2—combination of four random forest classifiers using the same feature set;
- *NN*—neural network using the same feature set.

Classifiers performance is shown in table 2a. We compute precision measure on negative examples to check whether our method is useful in eliminating erroneous and irrelevant contexts. We can notice that the results on English sides of examples are slightly better. This may be explained by the quality of word vectors which should be trained on larger corpus for languages with rich inflection.

Secondly, we use features for both sides to classify full examples. We apply random forest classification to the following feature sets:

- RF_3—*LM*, *MI, RelF* and *Sim* features;
- RF_4—all the features described in section 4.1.

Table 2b shows the evaluation results. The learning curves for RF_1 and RF_3 are presented on Figure 3.

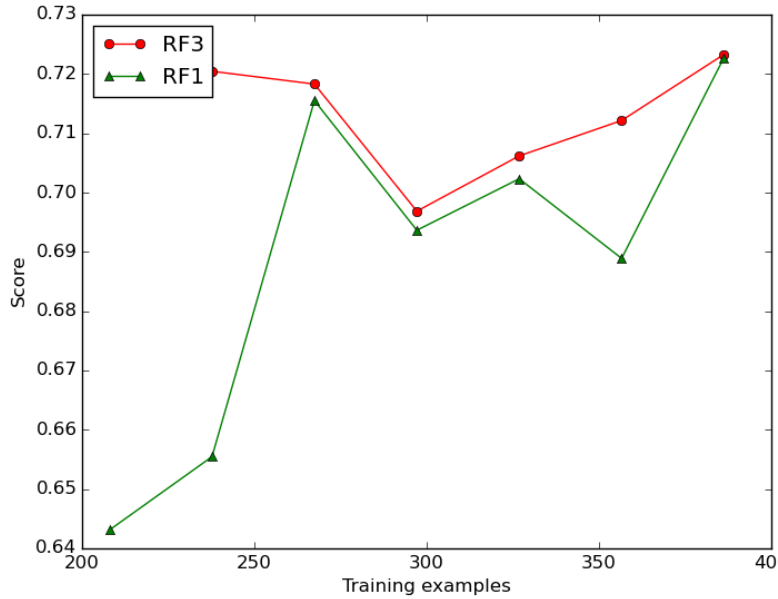

Fig. 3. Accuracy score for training sets of different size

5.2. Comparison with existing methods

Table 3. Number of correct examples extracted from different rankings

	correct examples	percentage of correct examples
MI	60	42.8
F	44	31.4
RF_1	59	42.1
RF_2	76	54.3
RF_3	**88**	**62.9**
RF_3	74	52.9

For comparative evaluation we choose 140 translation pairs, which were not annotated for the training set and extracted all possible context examples and selected top ones according to absolute example frequency *F* (i.e. f_1 in *RelF* formula from section 4.1) and *MI* metric described in Section 4.1. We compute MI for English and Russian phrase separately and then rank examples with respect to sum of scores for both sides.

We apply the same classification schemes to the resulting 22,375 examples and select the most relevant according to the following ranking:

- After RF_1, RF_3 and RF_4 classification we rank examples according to their scores (from 0 to 1).
- As mentioned before, the results of RF_2 classification include confidence scores for all values. When examples marked as good are found, we rank them according to their confidence score. When good examples appear only in one language, we select the pair with a positive value (4,5) and the highest confidence in one language and negative value (1,2) with the lowest confidence in another.

The results are shown in table 3. It can be observed that applying machine learning results in a noticeable improvement in examples quality. Examples acquired by RF_3 and selected according to the frequency ranking are compared in Table 4.

Table 4. Resulting examples, selected according to RF_3 and *F* scores

Key pair	RF_3	*F*
size—формат	standard size—стандартный формат	different sizes—различных форматов
control—контролирование	control costs—контролирование расходов	obstacle control—контролирование препятствий
guy—мужчина	white guy—белый мужчина	burly guy—дородный мужчина

Comparing different feature sets we can see that the most successful one is the one used by RF_3 classifier. These results in general correspond to those presented in Table 1 and Table 2, except for RF_2 classifier, which was expected to provide better results.

Taking into account feature importances computed by random forest we find out that the most important group is the *Sim* group. The direct comparison between word vectors (cosine similarity) seems to be the most relevant criterion, performing better than the internal vector comparison (when we use *WV* features). Using *WV* features with other groups proves to be redundant, although they would probably perform better on a larger training set.

The proposed confidence score improves the classification accuracy as compared with simple regression (RF_2 vs. RF_1). It would be interesting to apply this approach to classifiers with other groups of features.

6. Conclusion

We have described the procedure for automatically acquiring relevant illustrative translation examples for English-Russian machine dictionary. We have analyzed errors in phrases extracted from a parallel corpus in order to find out what features should be taken into account when choosing proper examples for a bilingual dictionary and discussed the drawbacks of straightforward approaches to ranking context examples. We have described our machine learning approach to detecting the most informative examples.

We have presented the results of classification and ranking evaluation. The comparison with simple methods proves that our approach overcomes such ranking functions as frequency or mutual information and may be successfully used for examples extraction. Some of the features proposed require minimal linguistic software so that the approach may be applied to other language pairs.

References

1. *Antonova, A., Misyurev, A.* (2012). Russian dependency parser SyntAutom at the DIALOGUE-2012 parser evaluation task. In Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog 2012" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2012"].
2. *Antonova, A., Misyurev, A.* (2014). Automatic Creation of Human-Oriented Translation Dictionaries. Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog 2014" [Komp'yuternaya Lingvistika I Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2014"].
3. *Bai, M.-H., Hsieh, Y.-M., Chen, K.-J., Chang, J.* (2012). DOMCAT: A Bilingual Concordancer for Domain-Specific Computer Assisted Translation. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. Jeju, Republic of Corea.

4. *Barlow, M.* (2004). Parallel Concordancing and Translation. Translating and the Computer.
5. *Kjaersgaard, P. S.* (1987). RefTex—a context-based translation aid. In D., Copenhagen University of Copenhagen (Ed.), Third conference of the European Chapter of the Association for Computational Linguistics: Proceedings of the conference.
6. *Koehn, P.* (2007). Moses: Open Source Toolkit for Statistical Machine Translation.
7. *Langlois, L.* (1996). Bilingual concordancers: a new tool for bilingual lexicographers. In Expanding MT horizons: Proceedings of the Second Conference of the Association for Machine Translation in the Americas. Montreal, Quebec, Canada.
8. *Mikolov, T., Chen, K., Corrado, G., Dean, J.* (2013). Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR.
9. *Och, F. J., Ney, H.* (2003). A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics, 29(1), 19–51.
10. *Wu, J.-C., Yeh, K. C., Chuang, T. C., Shei, W.-C., Chang, J. S.* (2003). TotalRecall: A Bilingual Concordance for Computer Assisted Translation and Language Learning. In ACL-2003: 41st Annual meeting of the Association for Computational Linguistics. Sapporo, Japan.