Categorizing Children
Automated Text Classification of CHILDES files

Rob Opsomer [a] Petr Knoth [b] Freek van Polen [c] Jantine Trapman [c]

Marco Wiering [d]

[a] *Ghent University, Faculty of Engineering*
[b] *The Open University, Knowledge Media Institute*
[c] *Utrecht University, Department of Philosophy*
[d] *University of Groningen, Department of Artificial Intelligence*

Abstract

In this paper we present the application of machine learning text classification methods to two tasks: categorization of children's speech in the CHILDES Database according to gender and age. Both tasks are binary. For age, we distinguish two age groups between the age of 1.9 and 3.0 years old. The boundary between the groups lies at the age of 2.4 which is both the mean and the median of the age in our data set. We show that the machine learning approach, based on a bag of words, can achieve much better results than features such as average utterance length or Type-Token Ratio, which are methods traditionally used by linguists. We have achieved 80.5% and 70.5% classification accuracy for the age and gender task respectively.

1 Introduction

In this paper, state-of-the-art text classification methods are applied to two tasks: categorization of transcribed children's speech according to gender and age. Various machine learning techniques from the field of text classification were applied. We summarize some widely used machine learning methods such as k-Nearest Neighbours, Neural Networks, Support Vector Machines and Boosting. We compare their performance on the two classification tasks. The text classification methods are based on the bag-of-words approach. In this approach, one looks at the frequency of words in a text, without considering their order.

The text classification methods are compared for their accuracy with traditional measures used in linguistics, such as the average utterance length and the Type-Token Ratio. Despite the fact that these measures are considered as standard measures, they are, as indicators of (morpho-)syntactic complexity, widely discussed for their reliability.

2 Data Set

Both the traditional measures and the machine learning methods are applied on a data set from the CHILDES (CHIld Language Data Exchange System) [1] Database [8]. This data set, named *Manchester*, contains speech of 12 different British children.

Every child was recorded approximately the same number of times, on average about 65 times. That way, we have a data set of 806 conversations. The distribution of the conversations over age and gender is quite equal. For the age classification task we have drawn the boundary at age of 2.4. This decimal number is both the mean and the median of the age in our data set.

Figure 1 shows the distribution of age and gender in our data set. The data set contains slightly more recordings from girls than boys.

[1] See `http://childes.psy.cmu.edu/`

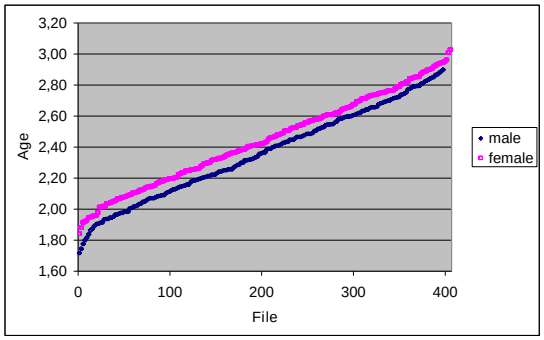

Figure 1: Manchester: female and male age distribution, in decimal years

2.1 Original document representation

In its original form, the Manchester data set is a highly structured data set, with all conversations transcribed in CHAT format for various linguistic layers. In general, each file contains two parts: a header and a body. The header contains metadata about the child, the other interlocutors, date of recording etc. The body consists of the transcription of the actual conversation. Apart from the utterances of the child, utterances of other interlocutors are included. Each utterance is displayed on a separate line and provided with an extra, morphological layer. In the example below the '*'-marked lines are utterances from mother and child respectively. The %mor-layers contain POS-tags of the words. The example shows also a typical transcription code: the part between brackets indicates that the word is guessed by the transcriber but omitted in the actual utterance.

*MOT: again ?
%mor: adv|again ?
CHI: where [0are] ?
%mor: adv:|where n|car-PL ?

There are a lot of such special transcription codes in the transcriptions. Examples are the use of 'xxx' and 'www' for unintelligible speech, and the insertion of the '+' character in compounds such as *ice+cream*. Even gestures and noises are included. All these codes are exhaustively described in the guidelines for the CHAT transcription: *Codes for the Human Analysis of Transcripts*.

2.2 Preprocessing

From the header the data about age and gender were extracted. All other metadata was discarded.

In the body, utterances from all other participants were discarded. Moreover, the morphological layers were also removed. Initially, we maintained two data sets, one with *plain* and one with morphological data. Since initial experiments showed inferior results with the morphological data, we have only continued research using the plain data set.

All special transcription codes were also discarded, and compounds were rewritten as single words (e.g. *icecream*). After preprocessing each file contained on average 616 words.

Original sentence:	**Processed sentence:**
CHI: <me go> [/] me go xxx .	me go

One of the stages within language acquisition is the differential phase which begins at the age of $2\frac{1}{2}$ years[2] [5]. In this phase a child starts to use inflected verbs (instead of infinitives), auxiliaries, modals and

[2]For some children this phase might start a little later

function words among others. The children involved in our project are at an age where this differential phase might start. This means that the use of suffixes and function words should improve the performance of the age classification. Hence, unlike regular text classification, stemming and removal of function words is *not* applied to our data.

3 Document Representation

After the initial preprocessing phase, the documents are converted into a form that can be used by the clasiffication algorithms. The documents are represented either by *naive features* or a bag-of-words.

3.1 Naive Features

The first conversion method is based on the extraction of a number of naive features from each document, such as the average sentence length or the amount of different words in the document. The latter can be normalised by the document length. In that case it is called the Type-Token Ratio. These measures are traditional methods used by linguists to acquire an indication of a child's language development.

3.2 Bag of words

The second way to represent documents is having each document represented by a feature vector of size n, where n is the number of entries in the dictionary. The values of the feature vector can be either 1) word frequencies denoting how often each word occurs in the document, 2) word frequencies as a percentage of the total number of words in the document, 3) binary values denoting for each word whether or not it occurs in the document, or 4) the more sophisticated term frequency-inverted document frequency (tf-idf) measure [9].

The dictionary consists of a number of words. The basic way to construct the dictionary, is to simply use all words that occur at least once in the training set. Since this generally yields a dictionary that is very large, a simple dimensionality reduction technique is applied. In this technique, only the words that occur at least τ times in the training set are included in the dictionary. Here τ is a threshold that can be varied at will. Experimental evidence suggests that using only the top 10 percent of the most frequent words does not reduce the performance of classifiers [3]. In our experiments, we have experimeted with the value of τ between 0 and 40.

4 Classification Algorithms

For the classification of our documents we use the following state-of-the-art methods for text classification [2]: k-Nearest Neighbours (k-NN), Support Vector Machines (SVMs), neural networks (NNs), and the boosting algorithm.

4.1 k-Nearest Neighbours

The k-NN method is easy and straightforward to implement, and it has been reported before that it is quite effective in text classification [9].

For classification of a certain document, the k documents that are most similar to this test document are selected from the training set. The categorization status value (csv) of the test document is then calculated as:

$$csv = \frac{1}{k} \sum_{i \in neighbours} csv_i \tag{1}$$

where the csv of a training document is always -1 or 1. The csv numbers represent a class: for example -1 is female, 1 represents male. A slightly modified variant for age classification:

$$csv = 1 - \frac{1}{k \times ageMean} \sum_{i \in neighbours} age_i \tag{2}$$

where the csv of a training document is always -1 or 1. For age classification the csv is a weighted function shown in equation (2). Since our classification tasks are binary, the csv is then rounded to either -1 or 1.

In order to determine which training documents are most similar to a test document, one needs a distance function. In this work, the Euclidean distance and Cosine similarity are used.

4.1.1 Cosine Similarity

The Cosine similarity is a common vector based measurement calculating the similarity between two vectors on a $[0, 1]$ scale. Similarity of 1 means for two vectors to be either identical or different by a constant factor. Given feature vectors $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$ the Cosine Similarity is defined as:

$$cos(X, Y) = \frac{x_1 \cdot x_2 + \cdots + x_n \cdot y_n}{\sqrt{x_1^2 + \cdots + x_n^2} \cdot \sqrt{y_1^2 + \cdots + y_n^2}} \tag{3}$$

4.2 Support Vector Machines

SVMs are very powerful classifiers for text classification [9]. The idea of SVM is based on simultaneous minimization of classification error and maximization of the geometric margin [10]. The original optimal hyperplane algorithm was proposed by Vladimir Vapnik in 1963 and involved a linear classifier. This was extended in 1992 applying the *kernel trick* to maximum-margin hyperplanes. Since that extension, the transformation may be non-linear in the original input space. Some common kernels are Radial Basis Function (RBF), Polynomial and Sigmoid.

In this project we use the *LIBSVM* package. LIBSVM is an integrated software package for support vector classification, regression and distribution estimation [1]. In this package, manually tuning of parameters is not necessary. By default, grid-search, which tries different combinations of parameters C and γ using cross-validation, is used to detect parameters that maximize accuracy. Note that only the RBF kernel is used in the experiments, because the linear kernel is a special case of RBF and the polynomial kernel has more numerical difficulties, since kernel values may go to infinity [6].

4.3 Neural Networks

In our work, we have used both linear (one-layer) and non-linear (multi-layer) neural networks.

In the linear NN, often called perceptron, the input layer consists of n input units, where n is equal to the amount of features each document is represented by. Each input unit is connected to one single output unit. The activation of the output unit is the weighted sum of the activations of the input units. Since the two classification tasks at hand are both binary, just the sign of the output unit suffices to determine which class it picked. The delta learning rule is applied to train the network. Learning is only performed when the network outputs the wrong sign, and if it does, it tunes the output toward values of 1 and -1.

In the multi-layer NN, there is also an input layer of size n, where n is equal to the amount of features each document is represented by. However, this time each input unit is connected to ten hidden units, whose activations are determined by a *sigmoid* function over the weighted sum of the input activations. These ten hidden units are connected to a single output unit. The activation of the output unit is the weighted sum of the activations of the hidden units. Backpropagation is performed to train the network. As with the linear network, learning (backpropagation) is only performed when the sign is incorrect.

4.4 Boosting

In the boosting method one iteratively trains multiple weak classifiers, each focusing on the prevention of mistakes the previously trained classifiers made. The actual classification task is done by combining the votes of the different classifiers and weighing them by the error they had on the training set. The boosting algorithm we used is called AdaBoost [4]. We have used the boosting algorithm in combination with linear neural networks.

	Gender	Age
k-NN	59.2%	76.2%
SVM	70.5%	80.5%
Linear NN	70.0%	74.6%
Naive features	50.0%	69.2%

Table 1: Accuracy results for the age and gender tasks

5 Experimental Results

All classifiers were extensively tested on a wide range of parameters. We have tried to determine the optimal parameters for all classifiers. The results can be found in table 1.

All results are presented as accuracy scores. Cross-validation is used in our experimental set-up. More specifically, a *leave-one-child-out* approach is applied. This means that one child is held out from the data set, and the classifier is trained on the rest of the documents. The resulting classifier is then tested on the held out child. This process is repeated for each child in the data set. The results are averaged over all children.

For the bag of words approach, results are discussed for the different classifiers. The naive feature results were obtained with the k-NN algorithm. All classifiers achieved very similar results with the naive features. If not mentioned, results describe bag of words experiments.

5.1 Gender Classification

The best results were obtained with SVMs, giving a score of 70.5%. A wide range of $\tau \in \{0, \ldots, 40\}$ produced similar results. This shows that for gender the most frequent used words are the most informative.

Linear NNs also performed quite good on this task. An optimal result of 70.0% was obtained using term frequencies, a learning rate of 0.0001, $\tau = 30$ and 200,000 randomly sampled training examples. However, note that this result is an average of eight trials. We noted some random fluctuations in the results of the networks. This is probably due to the fact that the training examples are sampled randomly, and that the weights of the network are initialized randomly. Also, the dimensionality of the task is quite high, while the amount of training data is rather small. However, it is unclear why this influences the results so much.

The non-linear NNs and the boosting algorithm were even more unstable, probably because of the more complex nature of these algorithms. Therefore, we don't include any results of these algorithms, although there were some promising results.

k-NN produced quite disappointing results for this task. A maximum score of 59.2% was obtained, with $k = 7$, no dimensionality reduction, tf-idf features and the cosine similarity function.

As expected, the naive features produced inferior results for the gender classification task. With accuracies around 50%, it didn't perform better than the random classifier.

5.2 Age Classification

Again, the best results were obtained using SVMs. The optimal result here is 80.5%, obtained with tf-idf features. A wide range of $\tau \in \{0, \ldots, 40\}$ produced similar results. This shows that for age the most frequent used words are the most informative.

With linear NNs, we obtained a best result of 74.6% with term frequencies, $\tau = 50$, a learning rate of 0.00005 and 500,000 randomly sampled training examples. Note that we averaged those results over 8 trials. Again, the non-linear NNs and the boosting algorithm produced very unstable results.

k-NN performed quite well on this task. An optimal result of 76.2% was obtained with $k = 3$, the modified csv computation (see section 4.1), $\tau = 2$, binary features and an Euclidean distance function. This result is very robust to changes in k and τ. The combination of tf-idf and cosine similarity produced also similar results.

With the naive features, an optimal result of 69.2% was obtained. Average sentence length (and not the Type-Token Ratio) seemed to be the most informative about the children age.

6 Conclusions and Discussion

Very good results were obtained with support vector machines. For the age task, those results are very robust across children. All children achieved accuracy higher than 60%. However, for the gender task, we have seen that boys are in general classified worse than girls. Some boys are even most of the time classified as girls. Similar observations hold for all classifiers. It would be very interesting to find out why this is the case.

When we look at the accuracies we have achieved (70.5% for gender, 80.5% for age), we can theorize what kind of implications they have. For instance, a 70.5% accuracy on gender can be interpreted as indicating that it is possible to determine whether speech comes from a boy or a girl by just looking at how often certain words are used. This observation supports the claim that boys and girls have different vocabularies already at such a young age. Also for age, an 80.5% accuracy indicates that it is very well possible to determine the age of a child only by looking at its speech (under the condition that the child is in the age stage where language acquisition plays a role). Though this is less remarkable than being able to determine gender, maybe it can be used to determine the rate at which a child is developing its cognitive skills. If a child is classified as being younger than it actually is, this may mean that the child is behind in language development.

When talking about current results, we should take into account that some conversations are very short (especially with the younger children), and are thus almost impossible to classify well. A possible solution would be to concatenate some of the conversation files. We should also take into account that some children are maybe very bad representatives of their class. For example when classifying children near the age pivot, this task can become almost impossible. For this reason, it would be interesting to make an experiment that would measure the accuracy of humans on our tasks. That would bring some real world comparison to the presented machine learning techniques.

A number of interesting things can be noted about the neural networks and the boosting algorithm. First off, most of the time each document is represented by a very large amount of features (1,500 or more). Because there are about 650 training documents every time, the neural networks are very likely to suffer from overfitting. Especially the multi-layer neural networks are very prone to overfit on the training data. This could be corrected by choosing a lot smaller feature set, in a more sophisticated way than just removing all words that do not occur more than a certain number of times.

What applies for the neural networks algorithm, also applies for the boosting algorithm, since we use linear NNs in the boosting algorithm. If the initial classifiers are trained too long, they will overfit and achieve (near-)100% accuracy on the training set. The succeeding classifiers will then focus on just very few examples, heavily overtraining on them. As a result of this, one will observe that the amount of samples from the training set for the boosting algorithm is much lower than the amount of samples for the networks. We observe that for the optimal setting we have found, changing the amount of classifiers reduces the accuracy. This indicates that the choice of the number of classifiers is very important especially when applying the boosting algorithm.

7 Future Research

For future research, there are quite a few daunting and promising options. The most promising one for the improvement of the results is probably to work on the features. For example, a selected set of function words or n-grams of parts of speech could be used. These features were already applied in [7], revealing distinctive differences between male and female (adult) authors. Some of our preliminary experiments have showed that using part-of-speech tags for the age task does produce good results. However, these results were never significantly better than the results obtained using the plain utterances.

A combination of plain and morphological data could also be interesting. One could also try a combination of children and parental speech. In order to do that, it should be established whether parental speech really gives a significant indication of the child's age or gender. There are already assesments from some of our preliminary experiments that it does.

Furthermore, it would be interesting to tackle the scarce data problem we have to deal with. More sophisticated dimensionality reduction techniques could be used, such as odds-ratio [9] or Latent Semantic Indexing. Another thing to consider is to search for more data. We tried the latter, but it proved to be fairly difficult to find another good data set.

Another thing that would probably be an easy way to improve our results for the gender classification task would be to drop the leave-one-child-out approach, and to adopt a leave-two-children-out approach. With the leave-one-child-out approach, the training data always consists of 6 children of the opposite gender of the child we want to classify and 5 children of the same gender, implying a bias for the opposite gender. For the age task, it might be interesting to cast the task as a regression task, instead of a classification task.

Finally, what might be interesting, is to discuss the sociological implications (discussed in the previous section) with linguists or specialists in the field of language acquisition of children. Perhaps, they have interesting ideas to improve the results. Or maybe, they can provide at least a theoretical foundation for some of the results we have obtained. Moreover, they could use the results of these experiments in their own theoretical and practical work. For this purpose, it would be interesting to inspect the words (or parts of speech) that distinguish between the classes.

References

[1] C.C. Chang and C.J. Lin. *LIBSVM: A Library for Support Vector Machines*, 2001.

[2] M. Cord and P. Cunningham, editors. *Machine Learning Techniques for Multimedia: Case Studies on Organization and Multimedia*, volume XVI of *Cognitive Technologies*. Springer Verlag, Berlin, 2008.

[3] R. Feldman and J. Sanger. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, New York, December 2006.

[4] Y. Freund and R.E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, September 1999.

[5] S. Gillis and A. Schaerlaekens. *Kindertaalverwerving: Een Handboek voor het Nederlands*. Martinus Nijhoff, Groningen, 2000.

[6] C.W. Hsu, C.C. Chang, and C.J. Lin. A practical guide to support vector classification. Technical report, Taipei, 2004.

[7] M. Koppel, S. Argamon, and A. Shimoni. Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, 17(3), 2003.

[8] B. MacWhinney. *The CHILDES Project: Tools for Analyzing Talk*. Lawrence Erlbaum Associates, Mahwah, NJ, third edition, 2000.

[9] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[10] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.