Predicting Liaison: an Example-Based Approach

Alexander Greefhorst — Antal van den Bosch

Centre for Language Studies, Radboud University, PO Box 90153, NL-5000 LE Nijmegen, the Netherlands

ABSTRACT. *Predicting liaison in French is a non-trivial problem to model. We compare a memory-based machine-learning algorithm with a rule-based baseline. The memory-based learner is trained to predict whether liaison occurs between two words on the basis of lexical, orthographic, morphosyntactic, and sociolinguistic features. Best performance is obtained using only a selection of lexical and syntactic features, yielding a best overall performance at a precision of .80, with recall at .85. Counter to our expectations, including sociolinguistic features even lowered the precision and recall of our predictions. The F-scores of the memory-based algorithm are higher than that of a simple baseline and three other state-of-the-art machine-learning algorithms. Based on the results on optional liaison, it appears that predicting liaison benefits from being able to generalize from specific examples in context.*

RÉSUMÉ. *La prédiction de la liaison en français est un problème de modélisation non trivial. Nous comparons un algorithme d'apprentissage automatique basé sur la mémoire avec un point de comparaison basé sur des règles. L'apprentissage automatique est entraîné à prédire si la liaison se produit entre deux mots consécutifs en évaluant des traits lexicaux, orthographiques, morphosyntaxiques et sociolinguistiques. Notre étude montre que la meilleure performance est obtenue en utilisant uniquement des traits lexicaux et syntaxiques, résultant en une précision de .80 et un rappel de .85. Contrairement à nos attentes, l'inclusion des traits sociolinguistiques rend la précision et le rappel plus bas. La F-mesure est la plus élevée en utilisant l'algorithme d'apprentissage automatique basé sur la mémoire. Elle est non seulement plus élevée que le point de comparaison basé sur des règles, mais aussi plus élevée que celle de trois autres algorithmes d'apprentissage automatique de pointe. Il paraît que la possibilité de généralisation des exemples spécifiques en contexte aide la prédiction de la liaison.*

KEYWORDS: *liaison; example-based models; memory-based learning.*

MOTS-CLÉS : *liaison ; apprentissage automatique basé sur la mémoire ; modèles basés sur des exemplaires.*

1. Introduction

Of all phenomena we can observe in the French language, liaison is among the more complicated. This is evidenced by the role of the phenomenon in French speech technology. Liaison is known as a source of errors in French text-to-speech systems (Yvon *et al.*, 1998; Pontes and Furui, 2010b). A missed liaison is an easily noticeable error, so a text-to-speech system for French should model liaison accurately (Béchet, 2001). Likewise, automated speech recognition systems for French should anticipate liaison in order to recognize words affected by it (Brousseau *et al.*, 1995). In both cases, the model would need to be as accurate as possible: it should predict liaison only where it would naturally occur (i.e. attain a high precision), and should not miss occurrences of liaison (i.e. attain a high recall).

Although many linguists have tried to model liaison already since the 16th century (Durand *et al.*, 2011), the debate has not arrived at a consensus. There are many special cases in which the practice deviates from the most obvious rules (Bybee, 2001), and there exist many cases in which liaison is neither obligatory nor prohibited, so it is hard to provide a correct and complete grammar for liaison. As grammar rules do not seem to fully explain liaison, perhaps rules are not the best starting point for modeling this phenomenon. Also, models that ignore frequencies of certain contexts triggering liaison miss some of the information (Bybee, 2001; Boula de Mareüil and Adda-Decker, 2002). It might be the case that native speakers do not (only) base their decision to make a liaison on a set of rules, but (also) on encountered similar cases and their frequencies. In that case, the predictions of a memory-based machine-learning algorithm might return results that rival those of a rule-based algorithm, or even outperform them. Therefore, we will try to answer the following question in this article: Is a memory-based machine-learning algorithm better in predicting the correct liaison consonant than a rule-based algorithm, in terms of precision and recall? These questions can be seen as a computational test of the hypothesis of Bybee (2001) that certain cases of liaison may better be predicted on the basis of similarity-based reasoning from memorized examples, which is what memory-based learning implements.

Thus, we have a double aim: to develop and test a computational model for liaison that is as accurate as possible, and to test a cognitively rooted hypothesis about how liaison would best be modelled computationally.

In order to achieve these aims, we will first briefly describe the phenomenon called liaison, in section 2. In section 3, recent studies on liaison are discussed. Then, in section 4, a brief introduction to memory-based machine learning is given. We describe our methods in section 5, our results in section 6 and a discussion of these results in section 7. In section 8, we state our conclusions.

2. Liaison

French possesses two (or three, depending on your definition) types of sandhi, processes that take place at word boundaries: linking (*enchaînement*) and elision/liaison

(Lodge, 1997; Eychenne, 2011). Linking causes coda consonants to be attached to the next syllable if this syllable has no onset, creating a resyllabified CV.CV. structure (*petite amie* “little friend (fem.)” being pronounced as [pə.ti.ta.mi] instead of [pə.tit.a.mi]). Elision is the process of deleting a word-final vowel (mostly a schwa) when the following word starts also with a vowel, as in the case of the definite articles (*le* and *la* becoming *l’*) and some personal pronouns (*ce est* becoming *c’est*). Liaison occurs in a similar context – when the next word starts with a vowel or a mute *h* – and is the process of pronouncing coda consonants that in other contexts would not be pronounced. Due to linking, which occurs in 99.2% of all liaison realizations in the PFC corpus, the liaison consonant becomes the onset of the following word. We can illustrate this with the following example: *grand père* [gʁãpɛʁ] (“big father”), where *grand* is pronounced without liaison, versus *grand ami* [gʁãtami] (“big friend (masc.)”), where the (normally not pronounced) final [t] emerges and is linked to the next syllable. We will not go into details of the phonological status of this [t] (i.e. is it latent or epenthetic); for discussions on this status we refer to Côté (2005) and Eychenne (2011).

Although there are many possible contexts for liaison to occur, it is not always realized. Three types of liaisons, each with its own historical roots (Laks, 2014), are distinguished in the literature: invariable (e.g. in *les _z_ arbres* “the trees”), variable (e.g. in *elle avait _t_ oublié* “she forgot”) and hypercorrect liaisons (e.g. in *sujet* | *intéressant* ‘’“interesting topic”). In grammar books, these three categories are also referred to as obligatory, optional, and prohibited (Lodge, 1997), where hypercorrect liaisons are violations of the prohibited condition.

In the 1960s, Schane (1968) proposed an analysis of liaison in only one rule, but this rule could not explain the differences between obligatory, optional, and prohibited liaison. One of the variables that seem to influence the realization of a liaison – the syntactic cohesion of a phrase – was already proposed by Delattre (1947). This cohesion is actually hard to represent in rules: when does a context have a sufficiently large cohesion? In *Le Bon Usage* (Grevisse and Goosse, 2011), fourteen syntax-based rules help to define whether liaison should be obligatory, optional, or prohibited, but their rules are prescriptions for formal language, and do not cover all contexts. Moreover, even if we can discover that a certain context belongs to the category “optional”, we still do not know whether the liaison will be realized or not.

The realization of a liaison however appears to depend on more than syntax only. Especially for the optional liaison, sociolinguistic factors are assumed to be of great importance. Lodge (1997) mentions the fact that realizing a liaison often is a marker of high status or prestige; grammar books will not say it is prohibited to realize optional liaisons in the *banlieue* of Paris, but they are hard to find there. Sociolinguistic factors such as age, gender, and conversation type are thus often included in analyses of liaison realization (Fougeron *et al.*, 2001; Adda-Decker *et al.*, 2012). Furthermore, if Bybee (2001) is correct in claiming that some liaison realizations are determined lexically, it might be worth the effort to include lexical properties such as word frequency and length. In section 5, we will look in more detail at the variables we included in our experiments.

3. Predicting liaison: related work

Not only linguists have trouble in formally modeling liaison in French – including computational linguists and speech technology researchers, in the domains of automatic speech recognition (ASR) (Brousseau *et al.*, 1995) and grapheme-phoneme conversion (Béchet, 2001) – but also children need some time to acquire it (Chevrot *et al.*, 2007). ASR developers and language-acquiring children encounter more or less the same initial problem: how to recognize a liaison. An illustration of this problem is found in the word *arbre* "tree": as children often hear *un arbre* [œ̃.naʁbʁ] "a tree", where the [n] is a liaison consonant, and they know that [œ̃] means *un*, they will initially think that [naʁbʁ] means "tree". This leads to children having several exemplars of one word in their memory, so for *arbre* they store /œ̃/+/naʁbʁ/ for *un arbre*, /le/+/zaʁbʁ/ for *les arbres*, /pəti/+/taʁbʁ/ for *petit arbre* (Chevrot *et al.*, 2009). At about age 5–6 years, children discover how and when liaison works, combine the different exemplars to one instance /aʁbʁ/, and stop making this error.

Several French ASR and grapheme-phoneme conversion systems have been fitted with liaison prediction modules (Brousseau *et al.*, 1995; Béchet, 2001). Although the effect of frequency on the presence of liaison is widely acknowledged (Bybee, 2001; Boula de Mareüil and Adda-Decker, 2002), most systems try to determine liaison by using rules that do not take this frequency into account. For instance, Boula de Mareüil *et al.* (2003) argue for about twenty rules to determine whether a liaison might occur or not. LIA_PHON, a grapheme-phoneme conversion tool of the computer laboratory of the University of Avignon (Béchet, 2001), refers back to the grammar rules of Grevisse (1993) in order to predict occurrences of liaison and account for those instances in the pronunciation.

Pontes and Furui (2010a) induce a model of liaison from data using a decision-tree learning algorithm. Trained on 1,500 liaison examples in context, extracted from the literature on the topic, they induce a system that predicts liaison on the basis of local features such as the part-of-speech of the word of which the final phoneme may undergo liaison, its final letters or phonemes, and the first letters or phonemes of the next word. Although they do not evaluate the prediction of liaison in itself, they report adequate improvement in a text-to-speech synthesis system.

Fougeron *et al.* (2001) study non-local linguistic and paralinguistic factors on the occurrence of liaison, such as speech style (spontaneous speech vs. reading aloud), general lexical statistics (average word frequency and length). Their overall conclusion is that these non-local factors do not sufficiently allow for the prediction of liaison. On the other hand, Adda-Decker *et al.* (2012) combined several recent studies and conclude that non-local factors should not be excluded since it is a "*phénomène multi-factoriel et multi-niveau largement influencé par des effets de fréquence*". In this contribution, we will not only rely on local features, but we will also investigate whether including non-local features can contribute to better results.

Our approach attempts to offer an alternative to rule-based approaches. In the next two sections, we will introduce memory-based learning, and discuss our methodology, including the corpus data and TiMBL, the software used in this experiment.

4. Memory-based learning

In order to test whether liaison can be explained by comparing a new instance with similar examples encountered before, we first explain the computational method we use for this test: memory-based learning, henceforth MBL (Daelemans and Van den Bosch, 2005). MBL does not work with a set of rules, but with a single memory filled with examples. When a memory-based system is asked to judge a new instance (e.g. to judge whether liaison should occur in a given context), it searches in its memory for the most similar instance(s), returning their (majority) prediction for the presence of liaison. The most important hyperparameter of MBL, which implements k-nearest neighbor classification (Aha *et al.*, 1991), is k, which determines the number of most similar instances sought.

As Daelemans and Van den Bosch (2005) point out, the similarity-based reasoning functionality of memory-based learning offers more than a mnemonic or rote learning capacity. Aside from matching newly encountered linguistic contexts with exact copies in memory, the memory-based learning approach also allows non-exact matches to base a decision on. More precisely, it will prefer exact matches, but will allow progressively non-exact matches up to the point where it has found k nearest neighbors among all memory instances.

Each instance in memory is represented by a feature vector. A feature can be anything: a number, a letter, a word, a phoneme, a part-of-speech tag, etc. When a memory-based system searches for similar instances, it searches for similar feature vectors, or in other words, it searches for vectors at a small mutual distance. This distance is the (weighted) sum of the normalized mutual distance for each feature. More formally, $\Delta(X, Y)$ as defined in Equation 1 is the distance between instances X and Y, represented by n features, and δ is the distance per feature.

$$\Delta(X, Y) = \sum_{i=1}^{n} \delta(x_i, y_i) \qquad [1]$$

For symbolic features, the distance function $\delta(x_i, y_i)$ is less obvious to compute than for numerical values (cf. Equation 2), but it is still possible. One solution is to attribute a distance of zero when two features are identical, and the highest possible distance when two features are different. As we use normalized values, this highest possible value will be 1. This solution, referred to as “weighted overlap” (Daelemans and Van den Bosch, 2005), is based on the IB1 algorithm as described by Aha *et al.* (1991), and is defined formally in Equation 2.

$$\delta(x_i, y_i) = \begin{cases} abs(\frac{x_i - y_i}{max_i - min_i}) & \text{if numeric, else} \\ 0 & \text{if } x_i = y_i \\ 1 & \text{if } x_i \neq y_i \end{cases} \qquad [2]$$

Another option is to use the modified value difference metric (MVDM), which takes into account the fact that some symbolic values are more similar to each other than others (Cost and Salzberg, 1993). In the example of liaison, if we would use letters as features, the difference between a *t* and a *d* could be considered smaller than the difference between a *t* and an *s*, because a word-final *d* and a word-final *t* can both lead to liaison in [t], while a word-final *s* often has liaison in [z]. As specified in Equation 3, the MVDM metric determines the similarity of the values of a feature by comparing co-occurrence of values with target classes. For the distance between two values $v_1,\ v_2$ of a feature, we compute the difference of the conditional distribution of the classes C_i for these values.

$$\delta(v_1, v_2) = \sum_{i=1}^{n} |P(C_i|v_1) - P(C_i|v_2)| \qquad [3]$$

For more possible distance metrics, we refer to Daelemans and Van den Bosch (2005).

Given a distance function able to compute the distance of a new instance to stored instances, the system can search for the k most similar instances, or nearest neighbors. The number of neighbors required to return the correct class varies for each situation. Daelemans and Van den Bosch (2005) have tested many possible numbers of neighbors, over six datasets, and their results show clearly that the optimal value for k has to be empirically estimated for each dataset on held-out data.

Thus, the optimal configuration of MBL algorithms differs unpredictably in each situation. Consequently, our experiment will not only concentrate on improving liaison prediction in comparison with a rule-based approach, but also on finding the optimal hyperparameter configuration.

5. Method

We first present the corpus and the software we used for our data. We then describe how we apply MBL to the data.

5.1. *Data*

The data we used in this experiment is derived from the PFC corpus, a corpus of contemporary French phonology[1] (Durand *et al.*, 2002; Durand *et al.*, 2009). In 2010, this corpus contained transcripts of 372 native speakers of French (not only in France, but also in francophone countries), who were recorded while reading a list of words and a text, being interviewed, and speaking freely (Durand *et al.*, 2011). For 205 speakers, dispersed over 18 *départements* in metropolitan France, each possible context for a liaison in the text and the two conversations was analyzed in Praat (Boersma, 2002) and coded by linguists and students, who not only had to indicate whether there was a liaison or not, but also in which consonant this liaison occurred. We used the interview and free speech components as data. As each participant is asked to read exactly the same word list and text, we excluded these recordings to avoid repetition in our data.

For each conversation, the first five minutes were coded. This means that, for each participant, we have ten minutes of speech, i.e. more than 30 hours of speech overall. This also includes the utterances of the interviewer and other conversation partners, which we excluded from our data. In the end, our corpus contained 291,208 words, between which in 5,444 cases (1.9%) a liaison is realized. The data contains 6,971 locations in which the rules of Grevisse and Goosse (2011) prescribe an obligatory liaison, of which only 4,220 (60.6%) are realized, and 856 locations in which the rules prescribe optional liaison, of which 288 (33.6%) are realized.

5.2. *Implementation*

The implementation of memory-based processing we used is TiMBL[2] (Daelemans *et al.*, 2010). This system allows us to use the configurations described in section 4, such as defining the way in which features are weighted and how the distance between vectors is modulated. Furthermore, it provides detailed precision and recall statistics and their harmonic mean, F-score (Van Rijsbergen, 1979).

5.3. *Data preprocessing*

In order to transform the PFC corpus into a format used by TiMBL, we extracted all words and liaisons encodings from the first five minutes of the Praat text grids. Where needed, text grids were edited manually to enable an automatic alignment. As described in section 2, lexical, syntactic, and sociolinguistic information may help to predict whether liaison occurs or not; therefore we collected also the age, gender, place

[1] The PFC data can be obtained from `http://www.projet-pfc.net`.

[2] TiMBL is an open source software (GPL v.3) and can be downloaded from `https://languagemachines.github.io/timbl/`.

of residence, job, activity (retired/(un)employed), education level, and the type of conversation (semi-formal interview or informal conversation with friends or relatives) of the speakers in the PFC corpus. The place of residence was coded as *département*; the job of a speaker was coded by the PFC into one of 22 discrete categories. Furthermore, we used TreeTagger (Schmid, 1994) to obtain part-of-speech tags of all words, and we used the rules described by Grevisse and Goosse (2011) to find out if and what type of liaison should occur in a specific context. These rules have not changed from 1993 to 2011, so even though we use a more recent edition of Grevisse, we could say we use the same rules as Béchet (2001). These rules are only taken into consideration when a vowel-initial word follows a consonant-final word. We refer to Appendix B for a description of our implementation of these rules.

Not all possible cases of liaison will be covered with the rules of Grevisse and Goosse; therefore we assign the category 'Not Found' if none of the rules above are applicable. This is the case for about 13,000 out of almost 25,000 instances (54.4%). The fact that if we would restrict our model to the predictions of Grevisse and Goosse (2011) we would miss 858 liaison realizations (6% of 13,000) in contexts that are not covered by these rules is a clear reason for seeking a computational model that would also generate predictions when the Grevisse rules have nothing to say.

We also included lexical information from the Lexique 3.8 database (New *et al.*, 2001; New, 2006) such as lexical frequency, number of syllables, and phonological distance. The lexical frequency metric is based on a large corpus of subtitles (New *et al.*, 2007). The phonological distance metric is based on the Levenshtein distance of the 20 nearest phonological neighbors (words that differ in only one phoneme), which is a better indication of phonological density than the number of neighbors itself (Yarkoni *et al.*, 2008).

Table 7 in Appendix A lists all features in more detail, including a description of how precisely they are encoded in our datasets.

5.4. *Feature selection*

Combining these sources of information allows us to create a richly coded data set. The question is whether all mentioned variables really contribute to the process of finding similar instances. Although the MBL algorithm is provided with a weight metric attributing lower feature weights to less predictive features, we ran a series of experiments to test if an incremental feature selection could yield higher results. We started with data sets containing the values of only one single features, used the default hyperparameter settings, and continued with the single feature yielding the highest results. Then we ran experiments with data sets consisting of this one feature plus only one of the other features. Step by step, we build a data set that only contains features that improve the results compared to the feature selection of the previous step, until adding a new feature does not improve performance. The dataset that turned out to return the highest results include the last five letters of the word(s) before the

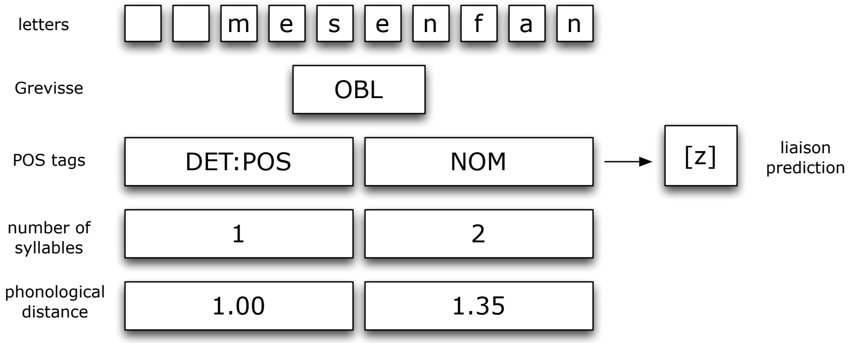

Figure 1. *Example features generated for the positive example of a liaison occurring between* mes *and* enfants *in the utterance fragment* Alors nous sommes allés à Montpellier puisque mes enfants sont à Montpellier, ...

possible liaison position and the first five letters after this position, the prediction made by the rules of Grevisse and Goosse (2011), the part-of-speech tags of the two words, the length of the words measured in syllables, and the phonological distance.

As TiMBL should be able to attribute low feature weights to non-contributing features, we performed our experiments twice: one time with all available features, and one time selecting only the most contributing features.

Figure 1 exemplifies the features generated for the liaison with the consonant [z] occurring between *mes* and *enfants* in the utterance context *Alors nous sommes allés à Montpellier puisque mes enfants sont à Montpellier, ...* Although rules predict that only the letters immediately surrounding the liaison position are relevant, we include a window of five letters to the left and right of the position. We also include the part-of-speech tags of the words *mes* and *enfants* along with their number of syllables and their phonological distance based on the Levenshtein distance to their 20 nearest phonological neighbors. Finally, we include the outcome of the rules of Grevisse and Goosse (2011) as detailed above, where “OBL” denotes the “obligatory” outcome of the first rule.

In order to verify whether TiMBL can predict the correct liaison consonant given a particular data set, we need to split the data into a training set and a test set. With 18-fold cross-validation on the 18 *départements*, we let TiMBL predict the liaison consonant of all instances from one single *département*; the other 17 are used as training material. In other words, the instances of a *département* were used once as test set and 17 times as training set. We believe a *département* subset represents a realistic, coherent batch of data. The average precision and recall scores computed per liaison consonant provide us with a realistic indication of how well liaison can be predicted with a memory-based learning algorithm when a new batch of data from a different *département* would be processed.

5.5. *Hyperparameter optimization*

Besides selecting the appropriate features, it is also important to select the best parameters for TiMBL. Selecting the wrong parameters, e.g. a too high value for the number of nearest neighbors, k, can lead to a wrong prediction by overgeneralization. In order to find the best hyperparameter values for TiMBL, we used a heuristic, progressive sampling search method (Van den Bosch, 2004). This algorithm starts by testing many of all possible configurations on a small subset of the train dataset (500 instances). The best configurations – based on accuracy, i.e. how often the right phoneme was predicted – will be tested again, but on a larger part of the dataset. From these results, again the best configurations will be tested on an even larger dataset, and this continues until there is only one best configuration left.

As in our original dataset about 98.1% of the instances did not occur any liaison, always predicting “no liaison” instead of any other consonant would lead to an accuracy of .98. As the recall of the other liaison categories might be affected by the high occurrences of this class, we tested if downsampling the “no liaison” category to a 1.9% would give better results. In this way, there was no difference anymore between the number of instances with liaison and instances without liaison. The instances that were excluded in this way were randomly chosen from all instances without liaison.[3] If TiMBL would now predict “no liaison” in all cases, the accuracy would be only .50. Downsampling will trigger the system to predict more liaisons than if we would use the entire dataset.

As our experiment comprises an 18-fold cross-validation setup, we apply the progressive sampling search algorithm to each fold. For each of the two datasets, we counted the most frequent settings, so we could test each fold of a particular dataset with the same settings. These most frequent settings are shown in table 1. Generally, we can observe that with both datasets a relatively high value of k (7 or 9) is optimal, combined with a metric that estimates a real-valued distance between symbolic values (MVDM) and a distance-weighting metric that gives a lower weight to more distant neighbors (by inverse-linear or inverse-distance). Detailed descriptions of these settings are available in Daelemans and Van den Bosch (2005).

Dataset	Features	Feature distance	Feature weighting	k nearest neighbors	Distance weighting
1	Best five	MVDM	InfoGain	9	Inverse Linear
2	All fourteen	MVDM	GainRatio	7	Inverse Distance

Table 1. *Hyperparameter settings selected for each of the two downsampled datasets, and the most frequently selected optimal hyperparameter settings.*

[3] If we were to exclude only the contexts other than consonant-final followed by vowel-initial, we might exclude hypercorrect liaisons, e.g. of the type *quatre yeux* [katʁ_z_jø] “four eyes”.

6. Results

The three consonants [z], [n] and [t] account for 99.5% of all liaison occurrences with 2,554, 2,026 and 837 occurrences in our data, respectively. In all tables, we will report on these three consonants in terms of precision, recall, and F-score (Van Rijsbergen, 1979) with $beta = 1$. F-score represents the harmonic mean of precision and recall. A prediction of a liaison is considered a true positive only when the speaker of the PFC corpus actually produced a liaison in this case. If not, even if the context were to be considered as optional liaison, it is counted as false positive. The scores of [z], [n] and [t] as well as the overall precision, recall and F-score on liaison prediction are shown in table 2. The averages are weighted for the different sample sizes of the folds (*départements*).

		[z]			[n]			[t]			overall		
Dataset	Features	pre	rec	F	pre	rec	F	pre	rec	F	pre	rec	F
1	Best five	.53	.98	.69	.61	.99	.75	.18	.95	.31	.43	.98	.59
2	All fourteen	.59	.98	.74	.64	.99	.78	.23	.94	.36	.48	.98	.65

Table 2. *Average results of MBL on downsampled data, with the core five features and all fourteen features (pre = precision; rec = recall; F = F-score).*

The precision scores obtained are generally low (.48 overall), while the recall scores are remarkably high (.98 overall). In other words, most of the cases in which there should have been a liaison were also predicted as such (high recall), but TiMBL overpredicted liaison, causing relatively many false predictions (low precision). Apparently, downsampling the data – filtering out instances without liaison until there are as many cases of liaison as negative cases – leads to overprediction. Therefore, we ran the same experiments once again, but without downsampling. This leads to the hyperparameters in table 3 and the results in table 4, which now exhibit a better balance between precision (overall .80) and recall (overall .85), leading to higher F-scores (overall .82).

Dataset	Features	Feature distance	Feature weighting	k nearest neighbors	Distance weighting
1	Best five	MVDM	InfoGain	15	Inverse Distance
2	All fourteen	Overlap	GainRatio	5	Inverse Distance

Table 3. *Hyperparameter settings selected for each of the two entire datasets, and the most frequently selected optimal hyperparameter settings.*

Finally, we ran a series of experiments using other machine-learning algorithms in order to compare them with MBL. These include IGTree[4], an algorithm that

[4] IGTree is implemented in the TiMBL package, `http://ilk.uvt.nl/timbl`.

		[z]			[n]			[t]			overall		
Dataset	Features	pre	rec	F	pre	rec	F	pre	rec	F	pre	rec	F
1	Best five	.83	.90	.86	.80	.94	.86	.67	.47	.55	.80	.85	.82
2	All fourteen	.83	.90	.86	.80	.92	.86	.59	.46	.52	.79	.84	.81

Table 4. *Average results of MBL with unsampled data, with the three core features and all nine features (pre = precision; rec = recall; F = F-score).*

builds decision trees (Daelemans *et al.*, 1997), Ripper, a rule induction algorithm (Cohen, 1995), and Naive Bayes, which makes predictions based on the conditional probabilities of classes and features, assuming independence between features (John and Langley, 1995). These three algorithms have shown in the past to be good predictors, but are more abstraction-driven than the pure memory-based algorithm, and thus can show us if a memory-based approach is really the best way to model liaison.

The hyperparameters of IGTree and Ripper algorithms were optimized progressively by the same hyperparameter optimization program we used for TiMBL. For IGTree, the only parameter that could possibly affect the results is the way the features are weighted; GainRatio turned out to be the most predictive weight metric. The best hyperparameters for Ripper were starting to write rules for the minority class first, allowing rules to cover as few as one instance and applying only one optimization pass. Naive Bayes has been tested in the Weka application (Hall *et al.*, 2009). In order to handle the numerical values, supervised discretization was used to convert these values into nominal ones, but this did not improve the precision and recall scores. Using a kernel distribution instead of a normal distribution did improve the results slightly, but its precision is still far below the other algorithms.

We also ran an experiment completely based on the rules of Grevisse and Goosse (2011), constituting a rule-based baseline system. This baseline predicts always liaison if the rules of Grevisse state either obligatory or optional liaison. Even though optional liaisons tend to be more often omitted than realized, always predicting optional liaisons increases, as we can see in table 5, both the precision and the recall of the [t]-phoneme – which is the phoneme that is the most associated with optionality. The results of these experiments are listed in table 6.

	[z]			[n]			[t]			overall		
Optional liaisons	pre	rec	F	pre	rec	F	pre	rec	F	pre	rec	F
Never predicted	.62	.80	.70	.72	.96	.82	.26	.25	.26	.61	.78	.68
Always predicted	.61	.80	.69	.72	.96	.82	.32	.59	.41	.58	.83	.68

Table 5. *Average results for the Grevisse baselines (pre = precision; rec = recall; F = F-score).*

	[z]			[n]			[t]			overall		
Algorithm	pre	rec	F	pre	rec	F	pre	rec	F	pre	rec	F
TiMBL	.83	.90	.86	.80	.94	.86	.67	.47	.55	.80	.85	.82
IGTree	.82	.87	.84	.80	.90	.85	.64	.47	.54	.79	.82	.80
Naive Bayes	.53	.96	.68	.74	.97	.84	.19	.92	.31	.44	.96	.61
Ripper	.82	.92	.87	.79	.96	.87	.67	.36	.47	.79	.85	.82
Baseline Grevisse	.61	.80	.69	.72	.96	.82	.32	.59	.41	.58	.83	.68

Table 6. *Average results for MBL, the decison-tree approximation IGTree, Naive Bayes, and Ripper, as well as the Grevisse baseline with optional liaisons always predicted (pre = precision; rec = recall; F = F-score).*

7. Discussion

We first turn our attention to the relation between selected features and the reported results. When we would use either words or letters as only features, without any other information, we would obtain already better (or at least more balanced) precision and recall scores than the rules of Grevisse we used as baseline. Using letters gives a slightly better result than using entire words, probably because if there is a liaison to be realized, the last letter of a word determines in which phoneme this liaison is, as the rules say. This information – which letter is the last one of a word – becomes implicit when using a whole word as feature. However, new data may contain words that have not been seen yet – hence the identity of the final letter is not implicitly available in those cases.

It is also interesting to look at the difference between the data sets with the core five features and with all fourteen features. If we look at the results of the downsampled datasets, we see that the results improve slightly by adding more data into the dataset: the recall remains .98, while precision increases from .43 to .48. In the original datasets, without downsampling, adding more data seems to be worse, as both precision and recall decrease by .01 when more features are selected. Especially the precision of the [t]-phoneme, which accounts for almost all optional liaison tokens, is lower if more features are added. If we now look at the features that have been selected among the best five, the only features we see are lexical (letters, number of syllables, phonological distance) or syntactical (part-of-speech tags, outcome of Grevisse). None of the sociolinguistic features have been selected, and if we would, they would even lower the precision and recall scores, in particular for the [t]. This is remarkable, since optional liaison is often in this phoneme, and sociolinguistic factors are, according to the literature, considered to be predictors of optional liaison.

The fact that the socio-economic feature containing the working field of the speaker was coded into 22 distinct categories may have played a role in this feature not being selected; the feature may have been more predictive when it were coded into only three categories. Still, we expected the other sociolinguistic factors such as age and gender to have at least some predictive strength; instead they only lower the precision and recall. Apparently, optional liaison cannot be explained by our sociolinguistic features, but only by lexical and syntactic factors.

This observation is supported by a quantitative analysis of the results. In our data, we observe 230 occurrences of the auxiliary *est* ("is.3SG.present") followed by a noun, adjective, or verb participle. When using only the core five features, TiMBL predicts 79 instances as liaison, a realization rate of 34.3%. In the PFC corpus, 80 instances are realized (34.8%); the realization rate of TiMBL is almost identical to the rate in the corpus. However, only 47 instances of the 79 are predicted correctly as liaison (true positives). The other 32 predictions of TiMBL are incorrect (false positives). This means that 33 actual realizations are missed (false negatives).

Adding the sociolinguistic features leads to 38 true positives, 42 instances incorrectly predicted as liaison, and 42 liaison realizations missed in the *est* cases. While the realization rate remains more or less the same, both precision and recall drop from .59 when using only lexical and syntactic information to .48 when including sociolinguistic information. In other words, it is hard for algorithms such as MBL to capture optional liaison, and it appears to become even harder when sociolinguistic information is taken into consideration.

The comparison displayed in table 6 allows us to compare the MBL algorithm we used with other prototypical machine-learning algorithms. The advantage of MBL over IGTree was to be expected, since decision trees are often a faster but less accurate way to predict linguistic data (Daelemans and Van den Bosch, 2005). The results of Naive Bayes are more surprising: the precision and recall scores seem to be more comparable to the downsampled MBL scores than to the scores when we used the entire dataset. Naive Bayes overgeneralizes the liaison classes: almost all cases in which liaison should have been predicted have been captured by Naive Bayes (recall of about .97), but in even more cases the prediction of Naive Bayes is incorrect (precision was less than .50).

Ripper is a serious competitor of the MBL algorithm. On the two most frequent liaison phonemes, [z] and [n], this rule induction algorithm even has a .01 higher F-score than the TiMBL algorithm. However, on the hardest task for all algorithms – the prediction of [t]-liaison with its high level of optionality –, Ripper does not retrieve as many liaisons as the MBL algorithm. Overall, it appears that the MBL algorithm yields the best performance.

8. Conclusion

We set out to study how accurately a memory-based algorithm could predict the presence of a liaison. To test this, we performed systematic experiments to define which features are required and which parameters should be used for the algorithm. Using words or letters as only features was already enough to outperform a rule-based baseline, and including other syntactic and lexical information gave high results in terms of precision, recall and F-score.

We expected sociolinguistic factors such as age, gender, employment, education level and conversation type to be important as well, but including these features did not lead to a higher precision or recall, and thus were not included in the final data sets. In the end, only lexical and syntactic factors can help to predict liaison.

Measuring the performance of our approach on the three most frequent liaison consonants, [z], [n], and [t], together accounting for 99.5% of all liaison cases in our data, we observe F-scores of .55 on [t], the least frequent of the three, and .86 both on [n] and [z] liaison. The best overall F-score is .82, with precision at .80 and recall at .85.

Using the output of the rules of Grevisse and Goosse (2011) directly resulted in a lower precision (.47) and a comparable recall (.80). These results are not surprising, as the rules of Grevisse and Goosse (2011) are assumed to cover only formal language. The low precision shows that, in contemporary spoken French, liaison is far less often realized than formal rules prescribe – which has been shown in many former studies. A low precision thus means that using the rule-based predictions of Grevisse and Goosse (2011) overgeneralize liaison realizations. A more sophisticated rule induction algorithm, Ripper, is not able to learn optional liaison as accurate as the MBL approach. A memory-based approach performs more or less similar to the rules of Ripper, but when focusing on optional liaison, which occurs most with [t]-liaison, the memory-based approach is a better predictor.

The possible combinations of parameters we can use with an MBL algorithm are virtually endless (e.g. the number of nearest neighbors to find can be any integer larger than zero) – and trying them all requires a large amount of time. Therefore, we used a progressive sampling search algorithm to find the best settings for the MBL algorithm. The best settings for our dataset were a modified feature value difference metric, using information gain to weight the features, and searching for 15 nearest neighbors. As each word boundary was used as one instance, most instances did not contain liaison. To avoid an imbalanced sample, we reduced the number of instances without liaison. Although this led to a high recall, the precision was very low (too many instances were incorrectly predicted as containing liaison). In other words, downsampling did not help us getting a better result.

What we may venture to conclude at this point is that our results confirm Bybee's (2001) hypothesis: memory-based learning, offering a computational implementation of exemplar-based processing, is able to predict liaison at least as accurately as a classic rule-based approach, and our results on [t]-liaison suggest that the memory-based approach is better in capturing optional liaison. Arguably, this is because the method is not only able to capture the regularities that the rules capture, but it is also able to capture implicitly the statistical preferences of certain contexts to trigger liaison that the rules cannot.

9. References

Adda-Decker M., Delais-Roussarie E., Fougeron C., Gendrot C., Lamel L., "La liaison dans la parole spontanée familière: explorations semi-automatiques de grands corpus", *JEP*, vol. 1, p. 545-552, 2012.

Aha D. W., Kibler D., Albert M., "Instance-based learning algorithms", *Machine Learning*, vol. 6, p. 37-66, 1991.

Béchet F., "LIA PHON: un système complet de phonétisation de textes", *Traitement automatique des langues*, vol. 42, n° 1, p. 47-67, 2001.

Boersma P., "Praat, a system for doing phonetics by computer", *Glot international*, vol. 5, n° 9/10, p. 341-345, 2002.

Boula de Mareüil P., Adda-Decker M., "Studying pronunciation variants in French by using alignment techniques", *Proceedings of the 7th International Conference on Spoken Language Processing*, p. 2273-2276, 2002.

Boula de Mareüil P., Adda-Decker M., Gendner V., "Liaisons in French: A corpus-based study using morpho-syntactic information", *Proc. of the 15th International Congress of Phonetic Sciences*, p. 1329-1332, 2003.

Brousseau J., Drouin C., Foster G. F., Isabelle P., Kuhn R., Normandin Y., Plamondon P., "French speech recognition in an automatic dictation system for translators: the transtalk project", *Proceedings of Eurospeech-1995*, 1995.

Bybee J., "Frequency effects on French liaison", *Typological Studies in Language*, vol. 45, p. 337-360, 2001.

Chevrot J.-P., Chabanal D., Dugua C., "Pour un modèle de l'acquisition des liaisons basé sur l'usage: trois études de cas", *Journal of French Language Studies*, vol. 17, n° 01, p. 103-128, 2007.

Chevrot J.-P., Dugua C., Fayol M., "Liaison acquisition, word segmentation and construction in French: a usage-based account", *Journal of Child Language*, vol. 36, n° 03, p. 557-596, 2009.

Cohen W., "Fast effective rule induction", *Proceedings of the 12th International Conference on Machine Learning*, Morgan Kaufmann, p. 115-123, 1995.

Cost S., Salzberg S., "A weighted nearest neighbour algorithm for learning with symbolic features", *Machine Learning*, vol. 10, p. 57-78, 1993.

Côté M.-H., "Le statut lexical des consonnes de liaison", *Langages*, vol. 158, p. 66-78, 2005.

Daelemans W., Van den Bosch A., *Memory-based Language Processing*, Cambridge University Press, Cambridge, UK, 2005.

Daelemans W., Van den Bosch A., Weijters A., "IGTree: using trees for compression and classification in lazy learning algorithms", *Artificial Intelligence Review*, vol. 11, p. 407-423, 1997.

Daelemans W., Zavrel J., Van der Sloot K., Van den Bosch A., TiMBL: Tilburg Memory Based Learner, version 6.3, Reference Guide, Technical Report n° ILK 10-01, ILK Research Group, Tilburg University, 2010.

Delattre P., "La liaison en français, tendances et classification", *The French Review*, vol. 21, n° 2, p. 148-157, 1947.

Durand J., Laks B., Calderone B., Tchobanov A., "Que savons-nous de la liaison aujourd'hui?", *Langue française*, vol. 169, p. 103-135, 2011.

Durand J., Laks B., Lyche C., "La phonologie du français contemporain (PFC): usages, variétés et structure", *in* C. Pusch, W. Raible (eds), *Romanistische Korpuslinguistik–Korpora und Gesprochene Sprache*, Gunter Narr Verlag, Tübingen, p. 93-106, 2002.

Durand J., Laks B., Lyche C., "Le projet PFC (phonologie du français contemporain): une source de données primaires structurées", *in* J. Durand, B. Laks, C. Lyche (eds), *Phonologie, variation et accents du français*, Hermès, Paris, p. 19-61, 2009.

Eychenne J., "La liaison en français et la théorie de l'optimalité", *Langue française*, vol. 169, p. 79-101, 2011.

Fougeron C., Goldman J.-P., Dart A., Guélat L., Jeager C., "Influence de facteurs stylistiques, syntaxiques et lexicaux sur la réalisation de la liaison en français", *Actes de TALN*, p. 173-182, 2001.

Grevisse M., *Le Bon Usage, Grammaire française, Refondue par André Goosse. Treizième édition revue*, Duculot, Paris/Louvain-la-Neuve, 1993.

Grevisse M., Goosse A., *Le Bon Usage, Grammaire française, 15e édition*, De Boeck, Bruxelles, 2011.

Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., Witten I. H., "The WEKA data mining software: an update", *ACM SIGKDD explorations newsletter*, vol. 11, n° 1, p. 10-18, 2009.

John G. H., Langley P., "Estimating continuous distributions in Bayesian classifiers", *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc., p. 338-345, 1995.

Laks B., "Diachronie de la liaison en français contemporain : le cas de la parole publique (1999-2011)", *in* J. Durand, K. Gjeerts, B. Laks (eds), *La Phonologie du français : normes, périphéries, modélisation*, p. 333-375, 2014.

Lodge R. A., *Exploring the French Language*, Arnold, London, 1997.

New B., "Lexique 3: Une nouvelle base de données lexicales", *Actes de la Conférence Traitement Automatique des Langues Naturelles (TALN 2006)*, 2006.

New B., Brysbaert M., Véronis J., Pallier C., "The use of film subtitles to estimate word frequencies", *Applied Psycholinguistics*, vol. 28, n° 04, p. 661-677, 2007.

New B., Pallier C., Ferrand L., Matos R., "Une base de données lexicales du français contemporain sur Internet: LEXIQUE™ // A lexical database for contemporary French: LEXIQUE™", *L'Année psychologique*, vol. 101, n° 3, p. 447-462, 2001.

Pontes J., Furui S., "Modeling liaison in French by using decision trees", *Proceedings of Interspeech-2010*, p. 186-189, 2010a.

Pontes J., Furui S., "Predicting the phonetic realizations of word-final consonants in context. A challenge for French grapheme-to-phoneme converters", *Speech Communication*, vol. 52, n° 10, p. 847-862, 2010b.

Schane S. A., *French Phonology and Morphology*, Massachusetts Institute of Technology, Cambrigde, 1968.

Schmid H., "Probabilistic Part-of-Speech Tagging Using Decision Trees", *Proceedings of International Conference on New Methods in Language Processing*, vol. 12, Citeseer, p. 44-49, 1994.

Van den Bosch A., “Wrapped progressive sampling search for optimizing learning algorithm parameters”, *in* R. Verbrugge, N. Taatgen, L. Schomaker (eds), *Proceedings of the 16th Belgian-Dutch Conference on Artificial Intelligence*, Groningen, The Netherlands, p. 219-226, 2004.

Van Rijsbergen C., *Information Retrieval*, Buttersworth, London, 1979.

Yarkoni T., Balota D., Yap M., “Moving beyond Coltheart’s N: A new measure of orthographic similarity”, *Psychonomic Bulletin & Review*, vol. 15, n° 5, p. 971-979, 2008.

Yvon F., De Mareüil P. B., Aubergé V., Bagein M., Bailly G., Béchet F., Foukia S., Goldman J.-F., Keller E., Pagel V. *et al.*, “Objective evaluation of grapheme to phoneme conversion for text-to-speech synthesis in French”, *Computer Speech & Language*, vol. 12, n° 4, p. 393-410, 1998.

Appendix A

Table 7 lists all features that have been used in the experiments, as well as the way how they were encoded in the datasets.

Variables	Encoding	Description
Words	String	Four features containing the two words preceding a context and the two words following it
Letters	Categorical	Ten features containing the last five letters of the preceding and first five of the following word
Age	Numerical	Age of the speaker
Département	Categorical	*Département* in which the recording was made
Gender	Categorical	Either male or female
Jobcode	Categorical	Working field coded in 22 discrete categories
Working status	Categorical	Either employed, unemployed or retired
Study level	Categorical	Highest educational degree of the participant
Style (register)	Categorical	Either interview or informal conversation
Grevisse	Categorical	Either obligatory, variable, prohibited, not found or not applicable
Syllables	Numerical	Number of syllables*
Phonological distance	Numerical	Levenshtein distance of the 20 nearest phonological words*
Frequency	Numerical	Frequency in the subtitle corpus*
Part-of-Speech tags	Categorical	Part-of-speech tag predicted by TreeTagger*

Table 7. *Description of each feature. Variables marked with an asterisk consist of two values: one for the word preceding the context, one for the word following it.*

Appendix B

Liaison is obligatory if it occurs:

– between the PoS tags DET or ADJ and NOM, NAM or ADJ (e.g. *les _z_ années*);

– between the PoS tag PRO:PER (unless the token *eux*) and the PoS-tag VER or the tokens *en* or *y* —- or vice versa (e.g. *nous _z_ avons*);

– after the tokens *non*, *pas*, *aucun*, *point*, *nullement*, *rien*, *jamais*, *plus*, *assez*, *autant*, *beaucoup*, *fort*, *moins*, *tant*, *tellement*, *tout*, *très* or *trop* (having as PoS tag ADV) (e.g. *tout _t_ entier*);

– after the tokens *chez*, *dans*, *dès*, *en*, *fors*, *hors*, *plein*, *rez*, *sans*, *sous*, *vers* (having as PoS tag PRP) (e.g. *sans _z_ atout*);

– in the construction *de* … *en* … (e.g. *de temps _z_ en temps*);

– in the construction … *à* … (e.g. *mot _t_ à mot*);

– in the lexicalized expressions *Champs Élysées*, *États-Unis*, *Nations unies*, *peut-être*, *Lot-et-Garonne*.

Liaison is prohibited if it occurs after:

– a singular noun: a token with PoS tag NOM, not ending in "s" or "x", except when the preceding token is not a plural determiner (i.e. it has the PoS tag DET but does not end on "s" or "x") or the number one (i.e. *un* or *une*) (e.g. *sujet* | *intéressant*);

– a plural noun as first part of a compound word: a token with PoS tag NOM, ending in "s" or "x", preceded by a plural determiner (i.e. it has the PoS tag DET and ends in "s" or "x"), a number that is not one, or the tokens *quelques* or *mêmes*), and followed by the tokens *à*, *au* or *aux*, and a token with as PoS tag NOM (e.g. *des moulins* | *à vent*);

– a verb in the second person, in the indicative or the subjunctive: a token with PoS-tag VER:pres or VER:subp, and a maximum token distance of two to the token *tu* (e.g. *tu chantes* | *agréablement*);

– the token *et* (e.g. *une pomme et* | *un abricot*).

Liaison is optional if it occurs:

– between the PoS tags VER and NOM or ADJ (e.g. *nous sommes _z_ heureux*);

– between an auxiliary in the third person (the tokens *est*, *était*, *fut*, *serait*, *soit*, *fût*, *avait*, *eut*, *aurait*, *ait* or *eût*, having as PoS tag VER) and a past participle (i.e. a token with as PoS-tag VER:pper) (e.g. *il est _t_ allé*);

– after the tokens *dont* and *quand* (e.g. *quand _t_ on voit*).