This is a preprint of an article accepted for publication in International Journal of Speech Technology, copyright © 2012. Please, cite as in the Journal:
http://www.springerlink.com/content/e702857569023407/

The Construction-Integration framework: A means to diminish bias in LSA-based Call Routing

Guillermo Jorge-Botana
Departamento de Psicología Evolutiva y de la Educación,
Facultad de Psicología,
Universidad Nacional de Educación a Distancia(UNED),
C/Juan del Rosal,10 (Ciudad Universitaria),
Madrid, Spain
e-mail: gdejorge # psi.uned.es

Ricardo Olmos
Departamento de Metodología de las Ciencias del Comportamiento,
Facultad de Psicología, Universidad Autónoma de Madrid,
Campus de Canto blanco, 28049 Madrid, Spain

Alejandro Barroso
PlusNet Solutions,
C/Albarracín,58. Local 12, 28037 Madrid,
Spain

Keywords: Call Routing, Call Steering, natural language, Latent Semantic Analysis, Construction-Integration, psycholinguistically motivated algorithms.

Abstract

Semantic technology is commonly used for two purposes in the field of IVR (Interactive Voice Response). The first is to correct the output of voice recognition devices based on coherence with a context. The second is to perform what is referred to as "call routing", requiring technology that categorizes utterances and returns a list of the most credible routes. Our paper focuses on the latter, aiming to use the Latent Semantic Analysis (LSA henceforth) computational model (Deerwester, Dumais, Furnas, Landauer and Harshman, 1990) together with the Construction-Integration model(C-I henceforth), a psycholinguistically motivated algorithm (Kintsch, 1998), to interpret, manage and successfully route user requests in an efficient and reliable manner. By efficient we mean that training is unnecessary when the destination model is altered, and exhaustive labeling of all utterances is not required, concentrating instead only on some sample destinations. By reliable we mean that the construction-integration algorithm attenuates the risks from intra-destination variability and word saliency. Technical and theoretical aspects are discussed. In addition, some destination assignment methods are tested and debated.

1. Introduction

The way IVR (Interactive Voice Response) is commonly used for call handling is well-known: there is a series of menus and fixed options organized into hierarchical levels that the user must navigate through before reaching the desired service. These procedures are presented verbally as menus (general at first, becoming more concrete at later levels). The user normally presses a key that identifies the desired option (using DTMF: Dual-Tone Multi-Frequency) or has to utter a short word that the system will recognize (using ASR: Automatic Speech Recognition). A second level is then reached, and so on until the true motive of the call is identified. One of the main limitations of this method is that the user is required to verbalize certain previously presented terms or expressions, and then navigate through part of the application to ascertain their need. There is also a problem of findability, as it is difficult if not impossible to incorporate all the functionality offered into the first level menus. As a result users often hang up, or waste a considerable amount of time when they are unable to find the real motive of their call among menu choices. Users are commonly irritated and complain about this kind of customer service system.

In response to these shortcomings and problems, innovative customer service procedures generally referred to as Call Routing are currently under development. These new systems all avoid restricted menus of possible words to verbalize. The customer is presented a more general question (for example: “Welcome to our service, how can we help?”), the aim being for the user to utter an open response that will be recognized, categorized and assigned correctly to a destination where a specialist agent will be waiting or an automatic mechanism will carry out the request.

These new systems require two phases: recognition and classification. (1) Firstly they must recognize the user's utterance. To recognize what was said, stochastic language models (SLMs) will be used. These calculate the probability of word occurrence (conditional probabilities based on words previously recognized in a word string). These grammars are not deterministic, but have rather been self-generated during processing of a corpus. In this way, n-gram models are constructed - normally 3-grams - and the probability of word fragments appearing is calculated and corrected. These

SLM models therefore need to be trained with large corpora. (2) Then what the user has said must be classified. For this purpose Natural Language Processing techniques are used. These are normally semantic models, either ontologies or statistical models of language such as vector space models - also extracted from large corpora - of which LSA is one example (for a review see Shi, 2008). This article is concerned with this second phase - classification.

2. LSA

There are several computational linguistic models for developing semantic technology to categorize texts, one of them being LSA, a well-established technique descended from earlier vector space models (Salton,1983). LSA was originally described by Deerwester, et al (1990) as a method for Information Retrieval. LSA had a certain advantage over other models, as it could be used to infer what a participant had in mind when querying a database. This led two authors, Landauer and Dumais (1997), to go beyond the original conception of LSA and reinvent it as a model of knowledge acquisition and representation. In their 1997 article, "A solution to Plato's problem", they outline the reasons why LSA might properly simulate the process by which humans learn word meanings. The associationist-empiricist philosophical school lay behind this conception, known in the new digital age as the connectionist school. Many authors have supported Landauer and Dumais's conception (Foltz, 1997; Kintsch, 2000, 2001, 2002; Quesada, 2008; Louwerse, 2008).

From that moment on it has been one of the most productive models in cognitive science for mathematically modeling cognitive language processes - for instance working memory (Kintsch, 1998; Kintsch & Patel, 1999) and comprehension of predications (Kintsch, 2001), sentences (Kintsch, 2008) and metaphors (Kintsch, 2000; Kintsch and Bowles, 2002), as well as resolving some problems in the language industry, for example developing web-based e-learning platforms (see Haley et al., 2005, 2007 for a long list).

LSA begins by analyzing a large linguistic corpus represented as an occurrence matrix (X). In this usually sparse matrix, a row represents a term and a column a context (normally a document, a paragraph or a sentence; in the case of call classification a destination cluster or call utterance). Each cell has the frequency with which each term occurs in each context. In this matrix the words that have a purely grammatical function such as prepositions, articles or pronouns, are normally eliminated. In any case, the frequency distribution of words means that some appear many more times than others. This asymmetrical distribution of terms means that once the occurrence matrix (X) is constructed, local (formula 1) and global (formula 2) weighting functions (entropy formulae) must be applied to it to obtain a weighting matrix (X_W). This phase is called preprocessing. The weighting functions transform each raw frequency cell x_{ij} of the matrix, using the product of a local term weight, l_{ij}, and a global term weight, g_i (formula 3). This process attempts to estimate the importance of a term in predicting the topic of contexts in which it appears (Nakov, Popova and Mateev, 2001), maximizing that of the terms that do not occur very often.

$$l_{ij} = \log(tf_{ij} + 1) \qquad \text{(1) Local weight}$$

$$g_i = 1 + \sum_j \left[\frac{p_{ij} \log(p_{ij})}{\log(n)} \right] \qquad \text{(2) Global weight}$$

Where:

$$p_{ij} = \frac{tf_{ij}}{gf_i}$$

tf_{ij} is the number of occurrences of term i in contexts j, gf_{i} is the total number of times term i occurs in all contexts, and n is the number of contexts.

$$X_{ij} = l_{ij} \cdot g_i \qquad \text{(3) Value of each cell}$$

After this comes the step which best defines LSA: the weighting matrix (X_W) is reduced via Singular Value Decomposition (SVD). With this technique the dimensionality of the original matrix (X_W) is reduced and a new matrix representing the latent semantic space is generated (Deerwester et al., 1990). This matrix represents the terms and contexts with a greatly reduced number of dimensions, so that the words or contexts have a relationship in the semantic space close to the way we semantically interpret them. With this reduced space semantic relationships emerge that might not be evident in the original matrix, so that SVD can be interpreted as a kind of noise elimination technique. The matrices resulting from the process, U_kS_k and S_kV_k' (figure 1) represent the terms and contexts respectively, with their dimensions reduced and weighted. The rows of matrix U_kS_k are the terms and the columns of matrix S_kV_k' are the contexts.

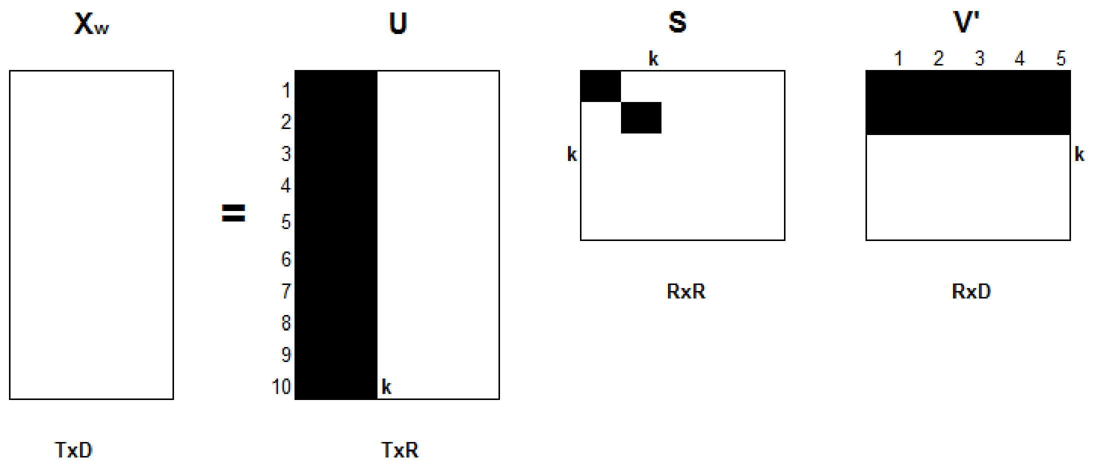


Figure 1: Reduced Singular value decomposition of the Term x Document matrix, Xw. k (≤R) is the chosen number of dimensions (factors) in the reduced model.

To assess the semantic relationship between two terms or two contexts (e.g. two utterances), the cosine between the two vectors is normally used (formula 4). A close semantic relationship between two words or two contexts is revealed by a high cosine, close to one in absolute value, whilst two words or contexts that are semantically unrelated produce a cosine close to zero (orthogonality). This is the essence of the LSA technique.

$$Cos(V_1,V_2) = \frac{V_1 \cdot V_2}{|V_1||V_2|} \qquad (4) \qquad \text{Similarity}$$

What relationship is there between this semantic model and call routing? LSA has mainly been used in two fields: correction of speech recognition hypotheses and assigning utterances to destinations (occasionally both are used in the same call routing application). Concerning correction of the speech recognition output, Jones & Martin (1997) have tested lists of possible common confusions made by speech recognition applications (homophones and near-homophones, {affect, effect}, {quiet, quite}), obtaining good results if such mistakes were corrected by checking the contextual coherence with indices of semantic similarity from an LSA model. These results are better than a Bayesian model, leading them to conclude that LSA is a good alternative to the Bayesian model under these conditions. In another study that links LSA with call routing, Bellegarda (2000) confirmed the usefulness of LSA in conjunction with a model based on *n*-grams. The authors adduce that the *n*-grams' analysis is limited to a very local context, and the concurrences analyzed by the LSA model greatly enrich the results of the router. Similarly Lim, Ma & Li (2005) obtained good results correcting speech recognition hypotheses and reassigning them new probabilities taking into account the coherence of the lexical context that accompanies each word. Shi (2008) also carried out a series of experiments to correct classification errors, correcting speech recognition outputs using indices of syntactic and semantic coherence (and a combination of the two). Shi reported that using LSA offered results comparable to those of Word-Net (although LSA was more economical and flexible), and that using certain parameters, a combination of the two measures led to an improvement in error correction. In an unpublished University of Memphis technical report, McCauley (unpublished) used LSA to define the deterministic grammars to be used in real time to recognize the responses to questions as they are formulated. If a user has mentioned Albert Einstein, the pertinent content words will be added to the rule - “theory of relativity”, for example.

As for assigning utterances to destinations, Chu-Carroll and Carpenter (1999) propose a system where an LSA module assigns utterances in response to the classic “say anything” to candidate destinations. If there is only one possible destination the utterance is routed directly to it. This will occur when the user's utterance vector and a vector that represents a destination have a high cosine (close to one), and the cosines of the utterance vector with other destinations are substantially lower. If the cosine between the request vector and destination vectors are all low, the call is transferred to an operator. Lastly, if there are several candidates, a disambiguation module is activated, seeking for the request to be reformulated. If the request vector is very close to both destination vectors, terms will be found that represent the difference vectors between the request vector and each of the destination vectors. Once found, only terms that can form bigrams or trigrams with some term from the original request will be used. For example, if a person says “loans, please”, this request might be assigned to two destinations: “loan service” or “customer lending”, given that both cosines are high. We need, then, to activate the disambiguation module. In this way, the difference vectors between “loans please”–“loan service” and “loans please”–“customer lending” will be extracted. With these vectors similar term vectors are sought that can form bigrams or trigrams with “loan”: let us suppose they are “car loan” and “loan payoff”. Using these words, questions are asked of users to disambiguate. One feature of Chu-Carroll and Carpenter's study is that in the LSA training phase they take 4,497 transcriptions from a banking services call-center. The occurrence matrix the LSA process begins with is formed by terms and routes (rather than terms and transcriptions), so that few columns are obtained - 23 to be exact. This is the criticism Cox and Shahshani (2001) level at Chu-Carroll and Carpenter: that they take the possible destinations rather than call transcriptions as contexts, so that LSA training is truly limited. They found better results in the laboratory if the contexts in the matrix are made from call transcriptions.

Li & Chou (2002) route calls via LSA introducing a variant at the preprocessing stage. When the corpus is trained, they calculate the Information Gain (IG) to identify the terms that really contribute information to the router. The IG index is based on variations in the entropy of documents in the absence and in the presence of the term analyzed. Later, Serafín and Di´Eugenio (2004) report good results using a variant of training where they introduce the labels the transcriptions were flagged with (the destinations or routes) as terms. In this way they managed to obtain cohesion between transcriptions routed to the same places. Also in this same year, Matula & Tyson (2004) obtained improvements introducing an additional step between recognition and routing of utterances. Instead of introducing the utterance from the voice recognition phase (with a generated SLM) directly, they correct it with the confidence indices provided. In this way, utterances of more than eight words (around 12 words) are routed slightly better, and this improvement is greater if the recognition software has more lax criteria (a lower confidence threshold).

As we have shown, the links between LSA and call routing have been quite carefully studied. Many of the LSA model's advantages can benefit call routing, but the representation of LSA contains some peculiarities that can add additional risks to the call routing process, and we should study them in some detail.

3. Predominant meaning inundation in LSA

Given the above, is clear that LSA could be a useful and efficient tool for assigning calls to destinations. The vector representation of terms has its advantages: allowing easy comparison between terms, as well as making similarities between words a probability-based issue rather than a dichotomy. We can establish a continuum of probability between a call and each different candidate destination using the cosine obtained between them. However, this vector representation hides some drawbacks that can only be avoided if the true nature of an LSA model is understood. An LSA model is a static representation of the terms, in other words, the LSA vector does not represent the true meaning of this term, but rather a means of storing an amalgam of its possible meanings. Although they might seem the same, the two concepts are different, since one thing is the meaning of a term in the real world, in other words, in interaction with a real context, and another thing is how this term is represented (in a system or even in the mind). LSA is only the latter. A term is devoid of meaning without a recall context - it takes on meaning as soon as it is verbalized together with other terms, or even accompanied by perceptual cues. With the representation alone we cannot deduce the exact meaning of a term. One of the phenomena where this becomes clear is polysemic words, and this helps us to understand the reality of other words, which whilst monosemic also have different nuances according to their lexical context and even the language community.

Even if a term has several meanings, it is represented by a single vector, constituting an average of the meanings this term has participated in, although the dimensions representing the most common contexts of occurrence will score higher. Given this representation, it would seem as though the vector represents a prototype that in reality does not exist, hence the paradox that the representation does not exist in the real world as such. This has led some authors to reflect that LSA accounts for synonymy but has difficulty representing polysemy (Deerwester et al, 1990).

Therefore, when it comes to mathematically extracting the content of this vector in the form of similarities with other terms, we sometimes find that one of the meanings does not appear; other times we find that the content is a somewhat arbitrary amalgam of meanings. Several previous studies attempt to show this dynamic based on visualizing the content of vectors in the presence or absence of trigger contexts (Jorge-Botana, león, Olmos & Hassan-Montero, 2010) and even in analyzing the justifications this type of representation might have in empirical data where humans evoke the meaning of the terms or perform tasks where no content is necessary (Jorge-Botana, León, Olmos & Escudero, 2011). In Jorge-Botana et al. (2010) a name is given to an effect that is confirmed when the meaning of terms is extracted: predominant meaning inundation. The only content often extracted is the predominant sense, the most frequent, since the others are not represented in the vectorial semantic space sufficiently to emerge in the form of meaning. This study indicates that if this is not the case, there is also a risk that the content extracted is totally imprecise and even occasionally aberrant, due to a stalemate between meanings. This second effect is referred to as imprecise definition. The fact is that the representation of an LSA vector is not directly translatable into the real world without bearing in mind a recall context and the way this constrains the meaning of each word.

Imagine a set of words forming the utterance “problems with a top-up card”. One of these words, “card”, has several meanings. The final interpretation of the utterance can occasionally be biased by how representative the different meanings of “card” are within the semantic space, if the other words are represented weakly or are very neutral. The interpretation may also be biased toward another meaning if “card” has another meaning that is predominant and which is not coherent with the lexical context (“problems with a top-up …”). This is the prototypical case: the word “card” is very common in the domain of phone services (“SIM card”, “pay-as-you-go card”, “top-up card”). This word is what we would call ambiguous. The problem is that if the word “card” occurs most often in one of these meanings - for instance “SIM card” - there will be a bias towards interpreting the utterance according to this predominant meaning, even when accompanied by a lexical context like “problems with a top-up” (which has a weak representation in the semantic space).

The technical explanation of such a case is as follows: To calculate the vector that represents the utterance, “problems with a top-up card”, LSA would normally calculate a new vector with a method known as folding-in[1]. The folding-in method projects the new utterance into the matrix of utterances[2] as an extra utterance, so “problems with a top-up card” would be introduced into the semantic space as a vector-utterance. However, this method is biased by the predominant meaning of the word “card”, because the final vector of the utterance is weighted by the value of each word it contains in each dimension: “card” scores high in one meaning, since it is represented in some dimensions which are predominant over other dimensions. This method is therefore dependent on the vector lengths of terms and their predominant meaning within the semantic space. To avoid this bias, we must bear in mind that “problems with top-up card” is more than the sum of the terms weighted by the value of the dimensions. The co-occurrence of these words constrains the meaning to terrain common to all and not toward the predominant sense of “card”. Since multiple meanings (or multiple nuances) of words are ubiquitous in language, this effect arises even in specific domains such as banking and telephony (see Jorge-Botana, Olmos & León, 2009 for an in-depth explanation using a domain-specific semantic space). In the following section we present the Construction-Integration model, a psycholinguistically motivated algorithm that can help to diminish this effect.

4. The Construction-Integration model

A technical report by the University of Memphis (McCauley, unpublished) contains the first and only mention of the possibility of using computational psycholinguistic models to correct this type of phenomena on an IVR platform. This author outlines an attempt to use a computational extension of the Construction-Integration model (see Kintsch, 1998). The importance that a network based on Construction-Integration might have for routers could be useful for modulating the words in an utterance to their correct meaning, taking into account all surrounding lexical context that constrains the final meaning. This is especially relevant for words that are ambiguous (such as “card”),

[1] Folding-In is specified later

[2] Matrix of contexts is generically known in Information Retrieval as matrix of documents. In order to be accurate in naming the context window, we prefer to call it “utterances”.

where the meaning that best matches the context will be given priority, thus avoiding the aforementioned predominant meaning inundation.

The basic assumption of computational models based on Construction-Integration is that interpretation of a text is carried out in two phases. In the first the content of each term in the text or sentence is evoked, and they are combined to build a network where each of them is joined to all the others. In the second phase, each of these terms receives activation from the others, proportional to the similarity they have with each other. After this phase, the most activated terms are those that belong to the main idea of the text (Kintsch, 1998). In this way a text is not seen as the sum of its constituent terms, but rather the integration of all of them into an end concept. Each term is constrained by the meaning of the others, and this generates the meaning. A variant of these models in a more local field is that which concerns the relationships of predicate structures. The content of a predicate is constrained by the meaning of the argument that accompanies it (Kintsch, 2001). This approach has even been used to model the comprehension of metaphors (Kintsch, 2000; Kintsch & Bowles, 2002). Recently, these models have been enriched with syntactic dependency relationships, allowing the meaning of complex phrases to be modeled (Kintsch, 2007). All the above models use LSA as a basis for building algorithms (although Latent Dirichlet Allocation (LDA), is beginning to be used). The network constructed weights its connections to reflect the similarity between each term and the others in the network, so that the activation levels each term receives depend on the similarity with the other terms. The meaning is constructed using a function that takes into account cosines taken from an LSA space.

Using the example above, to compute the vector of the entire utterance we do not need all the properties of each term (“problems” “top-up” “card”); we need only those that relate to the meaning of the whole context, and therefore extract the properties that are mutually activated by all the terms. Using this method, it is predicted that only the properties of cards related with “problems” and “top-up” will be selected. The full implementation procedure is described in the method section.

5. Aims

Of the two phases of Call Routing mentioned in the introduction (voice recognition and classification), this article is concerned with the semantic classification phase, taking utterances that are already transcribed and using the LSA model.

The main objective of this study was as follows:

1. Investigate the efficiency of C-I in correcting the predominant meaning inundation effect explained above, in LSA-based Call Routing. It is the first time that Call Routing with LSA explicitly incorporates such a mechanism, and we therefore need to ensure that a condition with C-I performs better in a routing task, with transcriptions of real utterances from a telecommunications service. We test this condition comparing it with a baseline: LSA direct routing.

2. Test several widely-used *cluster analysis methods to assign categories:* average, linkage and a hybrid method based on average and linkage. Both Direct Routing and C-I Routing methods will be tested with these three methods of category assigning.

3. Check the impact of labeling the training corpus on the efficiency of the router. One of the virtues of LSA is that in principle it requires no supervised training and does not need each utterance in the training corpus to be labeled. Whilst the labels give cohesion to the system (Serafín and Di´Eugenio, 2004), their absence might not lead to a substantial drop in efficiency of classification. If this is true, given the cost of manual labeling, transcriptions can be trained without labeling initially, or perhaps without any labeling at all.

These three aims will be explained in the method.

6. Method

The way LSA-based Call Routing assigns calls to a destination is by comparing the user's utterance (once transcribed) with some exemplars that represent destinations (also utterances); a destination might be represented by multiple exemplars. Both the destination exemplars and the utterance will be converted into vectors that can be interpreted by the LSA space. The effectiveness of the system will depend on the way destinations and utterances are converted into vectors and the way a destination is selected after comparisons between them. The following sections explain the means of doing so.

Training corpus: We began with a training corpus of 30,094 digitized utterances belonging to a phone company. LSA was trained with this corpus and the semantic space was created. These utterances were extracted using the Wizard-of-Oz procedure, and the transcriber assigned each of them a label that matched the destination they were routed to. It is important to note that the labeling was performed with different criteria as it was carried out at different times, so they are not truly exhaustive labels[3]. In any case, to give cohesion to the words related with one another, these labels were taken as an additional word, as in the Featured LSA used by Serafín and Di´Eugenio (2004). In other words, the LSA analysis finally covers the terms and labels that occur in utterances. We should also bear in mind that since one of the aims of this study is to check the efficiency of the system when LSA is trained without labels, in half of the experimental conditions we do away with them completely.

Training: In the LSA training we used Gallito®, a tool that has been used on other occasions for the creation of semantic spaces (Jorge-Botana, León, Olmos & Hassan-Montero, 2010; Jorge-Botana, León, Olmos & Escudero, 2010; Jorge-Botana et al., 2009). Words included in a special stop-list for this domain were eliminated, as well as all function words, and words that do not occur at least three times. In addition, some relevant words within the telephony corpus were artificially grouped into a single class - for example countries, mobile phone brands or phone companies, substituting them with the name of the class they belong to. We finally obtained a matrix of 1800 terms and 30924 utterances[4], although in the condition without labels we obtain a 1408 terms by 21667 matrix[5]. Before applying SVD and reducing the three resulting matrices to 270

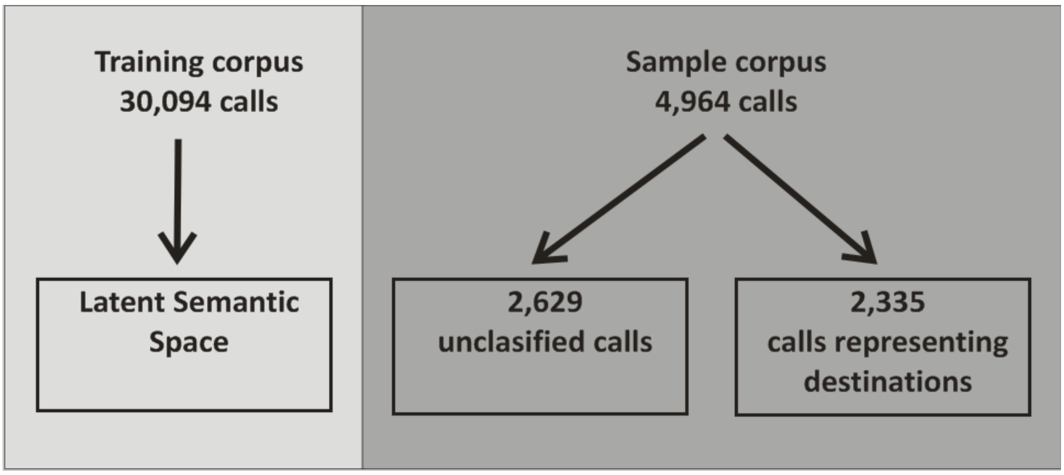
3 Strictly speaking, in LSA training these labels do not need to be exhaustive, so they can originate from different means of classification or even different corpora. In extreme cases, a sufficiently large corpus might even be trained without labels. Testing this last hypothesis is one of the aims of the present study.

4 The space is set up according to the method followed by Cox and Shahshani (2001), with a matrix built from terms and utterances, and not terms and grouped categories, like Chu-Carroll and Carpenter (1999). The rows of the occurrence matrix were terms (and also labels in the labeled condition) and the columns were utterances.

5 As well as having no labels, this decrease in the size of the matrix is due to single-term utterances (for example “credit”) being excluded from the training - undoubtedly this is a disadvantage of an LSA model without labels.

dimensions[6], log-Entropy is calculated, an estimate of the amount of information that a word carries in the documents where it appears. In this way, the terms that might contain most information are weighted more.

Destinations and test items: At the same time, 4,964 utterances were reserved to represent the destinations and to form the test items (see Figure 2). The sample set aside was randomly chosen, maintaining the proportion of calls assigned to each destination in the training corpus. Of the 4,964 reserved utterances, 2,335 were used to represent the destinations and 2,629 as test items (to launch them as calls and evaluate the reliability of the router at assigning them to destinations). Before the study, the labels for destinations that appeared in these 4,964 utterances were homogenized (the general corpus had destinations with much variability in naming standards). The task consisted of associating each destination from the 4,964 utterances set aside to one of the 29 final destinations which our final router model included[7]. Eventually, all of the 4,964 reserved utterances were exhaustively labeled with one of the 29 destinations. At the end of all this we have a semantic space of the telephony customer service domain, a set of utterances that are exemplars of the router model's 29 destinations and a set of test items.



6 We chose such a dimensionalization based on the assumptions made in some previous studies. In those studies it has been suggested that the optimal number of dimensions for specific domain corpora does not have to be extremely low, sometimes even approaching the 300 dimensions recommended by Landauer and Dumais (1997) for general domain corpora (See Jorge-Botana ,León, Olmos, & Escudero, 2010). Some of the most recent studies simply use 300 dimensions (Wild, Haley, & Bülow, 2011).

7 These models must cover all the functionality of the service.

Figure 2. Independent variables of the study: In line with the aims, to check efficiency at classifying calls with LSA we manipulated three variables for the analysis.

Independent Variables in the analysis: There are three independent variables that are manipulated when evaluating the reliability of the whole system.

(1) **Routing Method:** The first variable is how we represent the test utterances and utterances that represent the destinations vectorially in LSA. In this study two ways of vectorially representing each of the utterances are explored. One is Direct routing, where the utterances are projected onto the matrix of utterances ***V*** using the Folding-In method (formula 5). This is the classical technique, the way LSA has habitually been used (Manning and Schütze, 1999).

A new utterance-vector ***d*** can be created by computing an utterance ***c*** (a new vector column in the occurrence matrix X with all the terms that occur in it) and then multiplying it by $\boldsymbol{US^{-1}}$; ***c*** is also computed by applying the same global and local weights as in the creation of the original space.

$$d = c^T US^{-1} \qquad (5) \quad \text{Folding-in}$$

The second is C-I routing. The term draws on Kintsch's (1998) ideas as explained above[8]. In contrast with Direct Routing, C-I Routing has an intermediate step in the building of both destination exemplars and utterance (figure 3). This step is as follows. For each text, whether an utterance or destination exemplar, a network based on the Construction-Integration model is built (Figure 4). Firstly each term from the utterance

[8] Because call utterances are shorter and simpler than propositions within colloquial language, the algorithm which is used is not exactly the original construction-integration algorithm. The integration part proposed by Kintsch is a spreading activation algorithm which is iterative until the net is stable (the cycle when the change in the mean activation is lower than a parameterized value), whereas our algorithm is a "one-shot" mechanism. The activation of each node is calculated based on the connections received. Another difference with the CI algorithm as proposed by Kintsch is that we only consider words and not propositions nor situations. In any case, note that the original C-I is more complete and fine grained, but our mechanism is sufficient for our purposes and may be more flexibly programmed, because an OOP (Object Oriented Programming) paradigm has been used, with classes such as net, layer, node, connection, etc., instead of the iterative vector * matrix multiplication in the original (see Kintsch and Welsch, 1991 for details of the original conception).

is compared with all the terms present in the semantic space and the 50 most semantically similar neighbors to each are extracted. A connectionist network was created between all these neighbors (nodes of neighbors layer) and each of the original terms of the utterance (nodes of utterance layer), where the weights of each connection were given by the cosines[9] between each of the connected terms. Once the weights of the connections are assigned, the activation of each node is calculated based on the connections received. The more weight the connections received have, the greater the activation but the activation function also promotes nodes where the source of activation derives from several terms from the utterance and not just one or two (formula 6). The 20 most strongly activated neighbors are chosen, and a new utterance is constructed with them.

$$A_i = \delta_i \sum_{j=1}^{n} \log(1 + C_{ij}) \qquad \text{(6) Activation function}$$

Where ***j*** is the sub-index of the utterance layer, ***i*** is the sub-index of the neighbors layer, C_{ij} is the strength of the connection that node ***i*** received from node ***j*** (this last node belongs to the first layer), and δ is a corrector factor to avoid unilateral activation (based on the standard deviation of the connections received).

$$\delta_i = \frac{1}{\sqrt{\frac{\sum_{j=1}^{n} (C_{ij} - \overline{C}_i)^2}{n}} + 1}$$

> Where n is the total number of connections that is received by ***i*** (or the number of nodes in the neighbors layer) and C_i is the mean of all the strength that node ***i*** received

We might say that these terms produce a better definition of the text, an idea common to all words from the exemplars or from the utterances. In terms of its functionality, the procedure is analogous to the semantic common network algorithm (Olmos, León, Jorge-Botana & Escudero; 2009), implemented to improve classical methods of

---

[9]The cosines are calculated using the previously trained semantic space, in other words each of the terms to be compared is represented by a vector in this space. Any term vector might then be compared with another term vector using the cosine.

evaluating text with LSA, providing additional semantic information in the paragraph vectors.

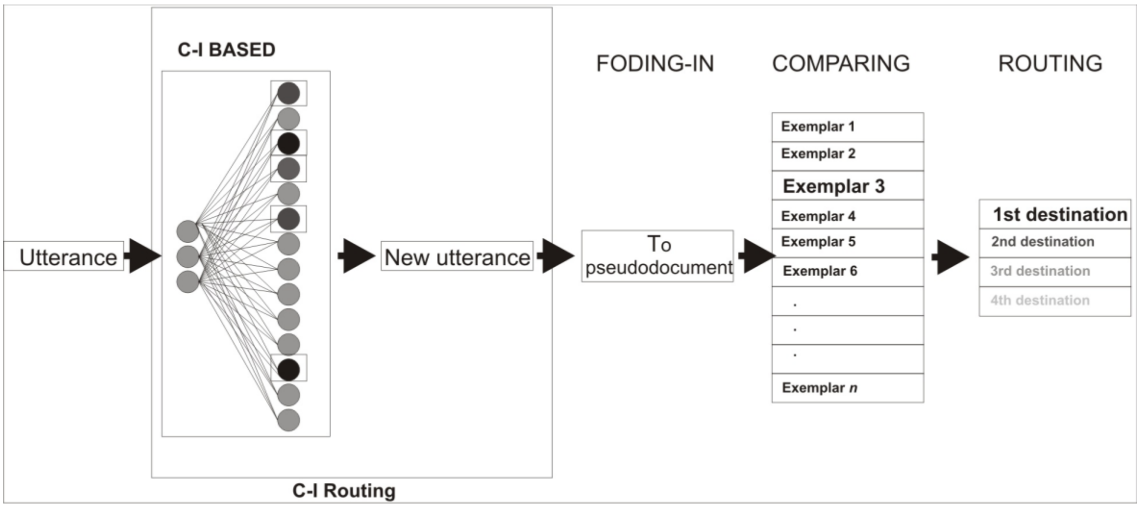


Figure 3. Graphical view of the call routing process for Direct routing and C-I routing

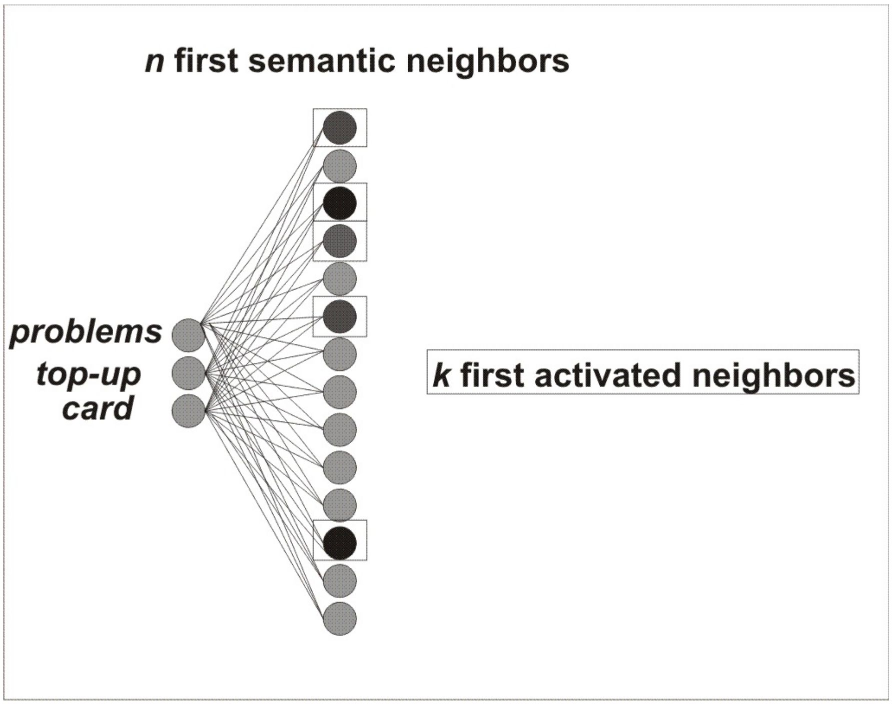


Figure 4. C-I based net implemented in this study. The K most strongly activated neighbors (with a square) will be represented in a pseudodocument.

This allows us to enrich the original utterance with relevant terms, adding terms that together properly represent the set of words that produce the call. Finally, this new enriched utterance is projected onto the latent semantic space, just as with *Direct routing. C-I Routing*, therefore, is an intermediate step that is introduced in the process of building both the destination exemplars and the utterances.

(2) **Method of category assigning:** The second independent variable manipulated in the study refers to assigning destinations**:** for this purpose two widely-used cluster analysis methods were used. Firstly, assignment to a destination was carried out using the *linkage method***,** consisting of the following: the utterance is compared with all exemplars. The selected destination is the one which has the most similar exemplar. The second destination in the list is the exemplar with the highest similarity (discarding the exemplars for the first selected destination), and so on. Alternatively, we use the so-called *average method*, whereby the selection of a destination is calculated with the highest average similarity within a destination. That is: the utterance is compared with each exemplar for a destination. The average of all the utterance-exemplar similarities for this destination is then calculated. This process is repeated for each destination and its exemplars. The selected destination is the one which has the highest average. The second is the one with the next highest, and so on. Lastly, knowing that both methods can provide different advantages we combine the two into a hybrid we call *average 4 method.* This method averages the four exemplars with the greatest cosine for each destination. The chosen destination is that whose average with the four exemplars is the highest. This method resembles Linkage in that it uses little information (Linkage uses the nearest exemplar), but it also resembles the Average in that it averages several exemplars.

(3) **Corpus**: The third independent variable manipulated refers to inclusion or otherwise of labels in the corpus: to check the impact of labeling the training corpus on the efficiency of the router, two corpora are formed, one with the source labels, described above, and another without these labels.

7. Procedure

There are, then, three independent variables that are manipulated when evaluating the reliability of LSA assigning calls to destinations: the method used to obtain the utterance vector (*Direct routing* and *C-I routing*), the method of category assigning (*linkage method, average method* and *average 4 method)* and whether the LSA space has been trained with a labeled or unlabeled corpus. Combining the three conditions with two, three and two levels respectively, we have a total of twelve ways to classify the utterances, and we evaluate the reliability obtained in each case.

The procedure could be summarized as follows: each of the 2,629 utterances from the set of test items is compared semantically (using the cosine) with the 2,335 destination representations, so that LSA assigns them to the most probable destination. As each utterance from the set of test items had already been assigned a correct ideal destination, coincidence between this and the destination assigned by the system was our source of data for measuring the effectiveness of each condition.

Lastly, we proposed a strategy for setting *thresholds* or cut-off points with each of these conditions, allowing us to decide when it is reliable to assign a call to a destination or when we must avoid doing so. Using logistic regression we can set more or less conservative cut-off points and conceive methods to foresee when it is best to disambiguate a call before assigning it to an operator. Two kinds of information are used in the logistic regression: the size of the cosine between the call and the destination, and the difference between this cosine and the cosine of the second most credible destination.

8. Results

a. Difference of averages for correct choices

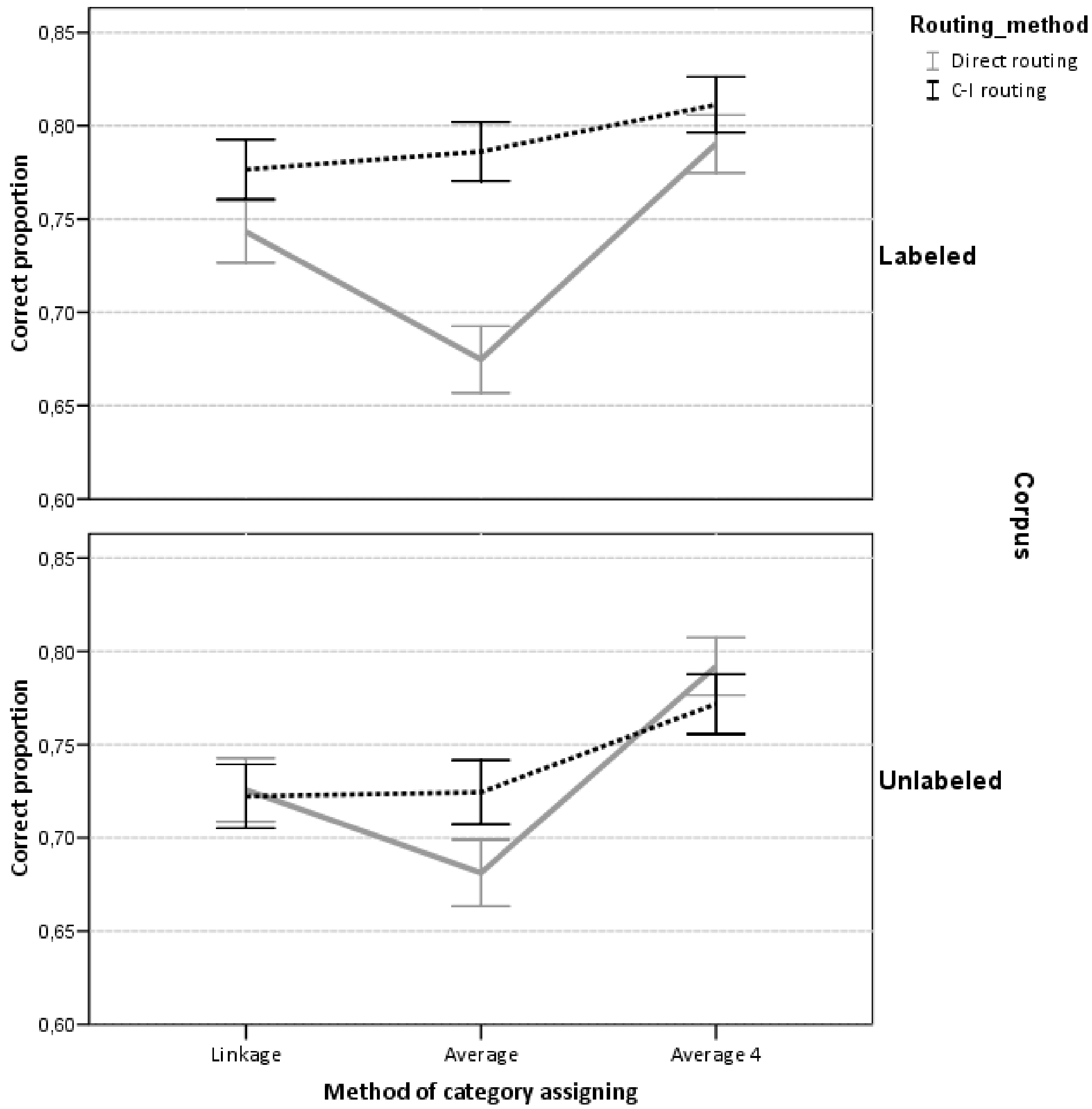


Figure 5. Triple interaction of the variables. Since the triple interaction was marginally significant and most within-pair comparisons broadly significant, this will be the reference graph.

We obtained a marginally significant effect for the triple interaction in a repeated measures ANOVA ($F(2,5916)=2.96$, $p=.054$), so the simple effects of this interaction were analyzed. Observing the triple interaction graph the interpretation of results is much simplified (Figure 5).

In terms of the method of category assigning (Average, Linkage and Average 4), we carried out the pair comparisons within each type of corpus (with and without labeling) and within the call routing method (Direct routing and C-I routing). Simple effects analysis showed that Average 4 obtained a significantly higher percentage of correct choices than the other two method of category assigning, even independently of the corpus and call routing method (*p* from .027 to .000). Therefore, Average 4 is shown to be a very robust method, and more efficient than the other two.

Secondly, we could say that overall the labeled corpus is best, as it obtained 2.8% more correct results, but depending on the condition there is a difference between the labeled and unlabeled corpus. Under the Average and C-I routing condition, there was a maximum difference between the two corpora of 6% (in favor of the labeled corpus, $p < .001$); it was minimal in the Average 4 and Direct Routing condition. In addition, the labeled corpus is the only one that benefits from the C-I method, since when the corpus was not labeled there are few differences between C-I routing and Direct routing. In fact, in Linkage there were no significant differences in favor of either, in Average C-I routing seems significantly more efficient ($p < .001$) and in Average 4 Direct routing gave the best results ($p < .01$). One of the causes of the better results for the labeled corpus is that it benefits from the effects of the C-I method, as if only Direct-Routing were used, the scores would be practically the same (see the V-shaped pattern for Direct Routing in both corpora)

Thirdly, and in the same line, C-I routing proved significantly more efficient than Direct routing in four of the six conditions with the other two independent variables. Thus C-I routing (see the graph) obtained a significantly greater percentage of correct results when the corpus was labeled. As we have already pointed out, there is a clear beneficial interaction between the labeled corpus and the C-I Routing method. When the corpus was not labeled, there were few differences between C-I routing and Direct routing. Overall, the benefit of the C-I method in the labeled corpus is not summative, as it is significantly greater when Linkage is used ($p < .01$) and especially in Average ($p < .001$), and lower using Average 4 ($p < .01$). The interpretation of this interaction will be discussed in the conclusions, but we will suggest that C-I is more beneficial in methods at risk from spurious word meanings or between destinations variability.

Other more practical conclusions are clear from the results of the ANOVA. Firstly there is no substantial loss in the percentage of correct choices for the best method, Average 4, when the corpus is not labeled. With labeling we reached 81% of correct choices with the best combination, Average 4 and C-I routing, and this fell only to 79% of correct choices in the best condition without labeling (in the Average 4 - Direct routing condition). This also indicates that Average 4 is a good choice regardless of other independent variables - it is a reliable method of classification. Secondly, we can point to Average 4 in combination with C-I routing, on a labeled corpus, as the best condition, since it is significantly better than all others. From a practical viewpoint, we should use this condition to obtain the best possible results.

b. Thresholds

To finalize our outline of the LSA call routing system, we need an equation that allows us to decide when to assign a call to a destination and when it is best not to do so. For this purpose we propose the logistic regression technique, given that it predicts dichotomous variables: in this case it predicts when routing will be successful or otherwise, according to the results obtained with LSA. At the same time, following on from the previous study we have selected the best of the conditions, combining the levels of the three independent variables as follows: Average 4, Labeled corpus and C-I routing.

To predict whether we should assign a call to a destination or not we use two covariables or predictor variables. One was the size of the cosine LSA awarded to the call, and the other was the difference between the cosines awarded to the first and second destination candidates. This second covariable was included since a greater difference implies less ambiguity for this destination, as the second destination is considerably less probable than the first.

After running the logistic regression model we were forced to use the second covariable (the difference between the first and second cosines) alone, given that there is too great a correlation with the first covariable (the size of the first cosine), making the model unstable (with a very low standard error due to multicollinearity). When we use the cosine difference the final model is as follows:

$$p(x=1) = \frac{1}{1+\exp^{.744-9.484*d\cos}}$$

which indicates that the probability of correct choices depends on *d cos* (the difference between the cosine of the first and second most credible options). The greater the difference between the cosines of the first and second options, the greater the probability that a correct choice is obtained, whilst differences between small cosines mean we cannot be sure of correct choices. In the classification table we can see that with this equation (and a conservative cut-off value of 0.5) decisions taken would be 85.2% correct. Out of 2,629 calls, 2,244 would be assigned to the destination proposed by the equation. Of those 2,244 calls, 1,994 proved correct (89% of the occasions where they were assigned). The model is conservative for 385 calls. These do not show a very large difference between the first and second cosine. Working in this way, we avoid 64% of classification errors. With the model there are 139 calls that would have been properly assigned, but that showed an insufficient advantage of the first over the second cosine (5% of all calls). Lastly, with the model there are 250 wrongly assigned calls (9.5% of all calls).

Classification table (a)

		Decisions after logistic model		
		Reject	**Accept**	**Percentage correct**
Results after logistic model	**Error**	246	250	49.6
	Correct	139	1994	93.5
Total		385	2,244	85.2

a Table 1. Classification table after logistic regression is run (The cut-off value is 0.500)

9. Conclusions

There are several major conclusions that can be drawn from the present study. Firstly we might focus on the economic aspect of a system like this in relation to the third aim of this study: analyzing the loss of effectiveness between labeled and unlabeled corpora. In one of the conditions the utterances from the corpus were labeled (although not exhaustively as labeling was carried out at different times and with different classification criteria, such that they often do not coincide). As the results have shown, LSA allows for the utterances of the corpus processed to not be labeled exhaustively (or even labeled at all). The results obtained with the labels are good, 81% in the best condition. Without labeling, the results are significantly worse, although there are conditions in which this loss is not too substantial (2% among the best conditions of both corpora). In general terms, then, we might say that labeling is best, but not labeling does not in any way imply an inefficient system when it comes to assigning calls to destinations.

In relation to this aim, we have observed that if a corpus is used without labeling, the C-I method does not provide the advantage that the labeled condition offers. Nonetheless, hardly any loss is found if the method is Direct Routing. This means that if only Direct Routing is used with both types of corpus (labeled and without labeling) very similar rates of correct choices are obtained. If we choose instead to reap the benefits of C-I, the unlabeled condition is unaffected by it. Our intuition tells us this might be due to C-I's reinterpretation of utterances containing labels, and these labels clearly representing the meaning - they will have greater vector length but the same low entropy (they are sufficiently representative words that bear information about the contexts they appear in). In this way, there is a lower risk that the words are poorly represented in the semantic space, compared with utterances that have not been enriched with labels. Ultimately, the labels are words that are well-represented in the semantic space (in terms of frequency and subject matter) and it is possible that for C-I to provide benefits a minimal representation of words from the space is required. We should remember that this space has been created using even words that occur only three times, so that in the absence of labels, many words will be spurious in terms of their contexts. It is probable that by increasing the number of training utterances and guaranteeing a minimum number of occurrences for each term, and therefore the representativeness of the words, the benefits of C-I without labeling will be reaped, although a large corpus is not always available so the opposite occurs.

One of the conclusions, then, is that although labeling (especially exhaustive labeling) leads to benefits, being more robust and revealing the benefits of C-I, it is not an essential condition, since the LSA algorithm will also capture the relationships between words. This in fact occurs in our study, where some of the conditions without labeling are not very far from the best labeled condition. In fact, except in specific call routing systems, few studies of LSA have used labels in the training of domain-specific corpora. In any case, we sense that the greater the training corpus, the less necessary the cohesion given by manual labeling and the advantage offered by labels over words in terms of representativeness. We might therefore say that the decrease in quality of labels and their heterogeneity provokes a grateful, non-decisive degradation in the system. This is an advantage compared to the extensive classification by some systems and their sensitivity to errors and inconsistency.

Continuing with labels, it is also important to highlight that in our system the only labels that have to be exhaustively labeled are the destination samples. We could say that this sample of destinations represents the intelligence of the system, and that they contain the representation of the business model - in this case of a telecommunications company. The efficiency of the system will be greater insofar as these destinations, which are also samples of utterances, correctly represent the functionality that needs to be differentiated and routed. One advantage is that the labels of these destinations can change or simply be classified without having to touch the general data structure - in other words without retraining and reclassifying the training corpus. This is an advantage we wish to highlight: adaptability to change and rearrangement of destinations. If we find ourselves needing to change the call routing model conceptually, we could do so by analyzing and varying the destination samples and their labeling.

Leaving pragmatics aside, and turning to purely methodological aspects, we would like to refer to the first of our aims: the call routing method. The results show the robustness of using the C-I method with LSA to represent utterances when the corpus is labeled. According to this data, C-I is a good way to attenuate the risks of intra-destination variability and word saliency. What C-I does is to reinterpret the content of an utterance and express it or even enrich it with other words (and labels). This is one way to attenuate the risk that one of the words in the utterance impregnates the whole meaning and occasionally biases the final meaning. For this reason, the positive effect of C-I is significantly greater in the conditions where these risks are higher such as Linkage and Average. Under the linkage method, the similarity with a destination utterance alone will be decisive, with the risk that this utterance is not the most orthodox destination sample. This can also be seen in the Average condition, where the average of the similarity with each utterance of the destination sample is calculated, with the risk of variability in the destination sample. This variability of destinations is almost inevitable, since there are destinations such as CALL OPTIONS that represent a broad spectrum of functionality (missed call alert, call diverting, collect calls, etc.) whilst for example LANGUAGES represents only a wish to be spoken to in a language. We say it is inevitable since the designers can be careful in the selection of destinations but can not avoid it entirely; in addition there is always the pressure that other departments wish to introduce definitions of destinations. All this seems to show that under some adverse conditions C-I might act as a protection, for example, when the destination sample is not fully refined (in fact, although it has not been reported here, we have been able to confirm that the benefit of the C-I method was more evident in preliminary, less refined versions of the destination sample). It also suggests that the reinterpretation of utterances in line with the lexical context (as in C-I Routing) protects from certain biases due to the word meanings, sometimes too closely tied to one meaning and other times imprecise. On the other hand, the fact that this protection does not exist in the unlabeled corpus might be because not labeling is an anomalous form of carrying out the process. On the one hand words are lost since single word utterances are not allowed (the matrix is considerably reduced), and also the words constituting reinterpretations are not always sufficiently significant or cohesive. In fact, we introduced this condition to check the negative effect on results and not to put our conditions to the test. As mentioned above, the considerable increase in the training sample and its possible causes in the unlabeled corpus is an area needing future research.

The third of the aims studied referred to the method of assigning a destination. The data leads to the conclusion that this should be based on the similarity of the caller's utterance with several utterances from the destination sample rather than just one of them (Linkage) as is normally the case. Additionally, it is not a good idea to do so using the average similarity with all the destination exemplars (as in the Average condition), but rather with a subset (as in Average 4) - those which most closely resemble the utterance introduced by the user. This is the method of assigning a destination that has proved most efficient, in the case of our study with the average of the four most similar utterances from each destination, although the number of utterances to consider might differ depending on the samples. We chose this amount as it proved outstanding in a series of informal tests.

The fourth feature to be covered is that applying acceptance thresholds using the logistic function has shown very good results. Chu-Carroll & Carpenter (1999) had already obtained better results if they applied a logistic function to predict the probability of correct choice given the similarity with the destination. In our system we have introduced another independent variable to the function, since as well as the similarity with the destination (or the average of similarities), we have used the difference in similarity between the first and second destinations on the list returned by the router. In other words, if the router returns four most probable destinations and their similarities, we introduce the difference between the similarity returned by the first destination and the similarity returned by the second. It is assumed that if this difference is large, the correct choice will be more probable, as the utterance has terms strongly tied to a destination. If it is small, the words will not be so clearly associated and there are greater risks. Taking these two variables, the similarity of the first destination (although this is withdrawn for offering lower variance) and the similarity of the first destination less that of the second, results are greatly improved. The cost is to make some individuals repeat their request, perhaps asking for more clarity (this type of strategy is part of the workflow design and the definitions of the texts to be used). It the end, the percentage of correct choice rises to 85% in the best scenario, meaning the router can obtain values that can be reliably implemented without excessive economic costs.

10. Future research

There are three fundamental issues that arose spontaneously from this research and that call for further examination in future studies. The first is to know how a system like this, with C-I and thresholds of acceptance, behaves when it receives utterances that are not transcriptions but rather from a voice recognition application. Our intuition is that the C-I model will be used to dilute the effect of confusions (insertions and substitutions, etc.) that the recognition module introduces into utterances. The justification for this reasoning is as follows. Given that C-I reinterprets the utterance based on the semantic context, it is possible that some utterances may benefit from the effect by which an error from the recognition stage has phonetic but not semantic similarity. The errors will be words outside the general context of the utterance, and receive little activation from the other words, so the errors lose part of their negative effect. An utterance with errors also receives less confidence from the logistic function.

Another line is that which concerns the list of most probable destinations and their treatment. The router returned a list of four destinations in descending order according to their similarity, as well as a confidence level extracted from the logistic function. When this confidence indicator exceeds a threshold, the first destination is accepted. This is the current method. If the threshold is 0.4 and the confidence level is 0.3, this destination is not accepted; if it is 0.5 it is, and it is routed. An alternative method is to divide the acceptance into three zones delimited by two thresholds. One is higher, above which the destination is accepted directly and the call is routed to this destination, and the other lower below which the destination is rejected and another attempt is suggested. Lastly, the confidence interval between the two thresholds passes to a disambiguation module in which the list of four destinations is used to suggest menu options. According to our preliminary calculations, we would obtain a 91% probability of correct choices using the first three destinations. Again this is a workflow design issue whose implementation might be very effective.

In addition, as we have already pointed out in the conclusions, the effect of the corpus size on the interaction we have detected remains to be seen, under the hypothesis that the greater the linguistic corpus for training LSA, the lesser the effect of exhaustive labeling on the percentage of correct choices. The more material used to train LSA, the better represented the terms in the semantic space, so that additional help, such as the labels that contribute coherence to the semantic space, will be increasingly unnecessary. In this sense, by controlling for corpus size we might analyze the benefits of labeling and not labeling the corpus, and the contribution that C-I routing offers on these corpora in relation with Direct routing.

References:

Bellegarda, J.R., 2000b. Exploiting latent semantic information in statistical language modeling. *In Proceedings of the IEEE*. 88 (8), 1279–1296.

Boon Pang Lim, Bin Ma, and Haizhou Li, "Using Semantic Context to Improve Voice Keyword Mining", *In Proceedings of the International Conference on Chinese Computing (ICCC 2005)*, 21-23 March 2005, Singapore.

Chu-Carroll,J., y Carpenter,B.,(1999). Vector-based natural language call routing, *Computational Linguistics*, v.25 n.3, p.361-388.

Cox, S. And Shahshahani, B. (2001). A Comparison of some Different Techniques for Vector Based Call-Routing. *In Proceedings of 7th European Conf. on Speech Communication and Technology*, Aalborg.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing By Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41, 391-407.

Foltz, P. W. (1996). Latent semantic analysis for text-based research, *Behavior Research Methods, Instruments and Computers*, Vol. 28, No. 2, pp.197–202.

Haley, D. T., Thomas, P., De Roeck, A., & Petre, M. (2005). A research taxonomy for Latent semantic analysis based educational applications. Technical Report no. 2005/09. Open University.

Haley, D. T., Thomas, P., Petre, P., & De Roeck, A. (2007). Seeing the whole picture: Comparing computer assisted assessment systems using LSA-based systems as an example. Technical Report Number 2007/07. Open University.

Hu, X., Cai, Z., Wiemer-Hastings, P., Graesser, A. C., & McNamara,D. S. (2007). Strengths, limitations, and extensions of LSA. In T. K. Landauer, D. S. McNamara, S. Dennis, & W. Kintsch (Eds.), *The handbook of latent semantic analysis* (pp. 401-426). Mahwah, NJ: Erlbaum.

Jones, M.P., Martin, J.H.: Contextual spelling correction using latent semantic analysis. *In Proceedings of the Fifth Conference on Applied Natural Language Processing*. pp. 163-176. (1997)

Jorge-Botana, G., Olmos, R., León J.A. (2009). Using LSA and the predication algorithm to improve extraction of meanings from a diagnostic corpus. *Spanish Journal of Psychology*. Vol. 12, No. 2, 424-440.

Jorge-Botana, G., León, J.A.,Olmos, R., & Escudero, I. (2010). Latent Semantic Analysis Parameters for Essay Evaluation using Small-Scale Corpora. *Journal of Quantitative Linguistics*, Vol 17, Number 1, pp. 1–29

Jorge-Botana, G., León, J.A.,Olmos, R. & Hassan-Montero, Y.(2010). Visualizing polysemy using LSA and the predication algorithm. *Journal of the American Society for Information Science and Technology*. Vol 61, Issue 8, pp. 1706–1724.

Jorge-Botana, G., León, J.A., Olmos, R., & Escudero, I. (2011). The representation of polysemy through vectors: some building blocks for constructing models and applications with LSA. *International Journal of Continuing Engineering education and Long Learning*, Vol. 21, No. 4.

Kintsch, W. (1998). The representation of Knowledge in Minds and Machines . *International Journal of Psychology*, 33-6, 411-420.

Kintsch, W. (2000). Metaphor comprehension: A computational theory. *Psychonomic Bulletin & Review*, 7, 257-266.

Kintsch, W. (2001). Predication. *Cognitive Science*, 25, 173-202.

Kintsch, W. (2007). Meaning in context. In Landauer, T. K, McNamara, D., Dennis, S. & Kintsch, W. (Eds.) Handbook of Latent Semantic Analysis. Mahwah, NJ: Erlbaum. pp. 89-105.

Kintsch, W., & Bowles, A. (2002). Metaphor comprehension: What makes a metaphor difficult to understand? *Metaphor and Symbol*, 17, 249-262.

Kintsch, W., & Welsch, D. (1991). The construction-integration model: A framework for studying memory for text. In W.E. Hockley & S. Lewandowsky (Eds.), Relating theory and data: Essays on human memory in honor of Bennet B. Murdock (pp. 367-385). Hillsdale, NJ: Erlbaum.

Kintsch, W., Patel, V., Ericsson, K. A. (1999). The role of Long-term working memory in text comprehension. Psychologia, 42, 186-198.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: the Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104, 211-240.

Li, L. & Chou, W. (2002). Improving latent semantic indexing based classifier with information gain", *In Proceedings of the 7th International Conference on Spoken Language Processing, ICSLP-2002*, 1141-1144. September 16-20, 2002 Denver, Colorado, USA.

Lim, B.P, Ma, B., and Li, H, "Using Semantic Context to Improve Voice Keyword Mining", *In Proceedings of the International Conference on Chinese Computing (ICCC 2005)*, 21-23 March 2005, Singapore

Louwerse, M.M. (2008). Embodied representations are encoded in language. *Psychonomic Bulletin and Review*, 15, 838-844.

Manning, C.D. and H. Schütze. (1999) *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, MA.

McCauley, L. (unpublished). Using Latent Semantic Analysis to Aid Speech Recognition and Understanding. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.3694

Nakov, P., Popova, A., & Mateev, P. (2001). Weight functions impact on LSA performance. *In Proceedings of the Recent Advances in Natural Language Processing conference – RANLP 2001*, Tzigov Chark, Bulgaria.

Olmos, R., León, J.A., Jorge-Botana, G. & Escudero, I. (2009). New algorithms assessing short summaries in expository texts using Latent Semantic Analysis. *Behavior Research Methods*. 41 (3), 944-950

Salton, G. & McGill, M. J. (1983). Introduction to Modern Information Retrieval. McGrawHill, New York, NY.

Serafin, R. & Di Eugenio. B., FLSA: Extending Latent Semantic Analysis with features for dialogue act classification. *In Proceedings of ACL04, 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain, July.

Shi, Y., 2008. *An Investigation of Linguistic Information for Speech Recognition Error Detection*, Ph.D. University of Maryland, Baltimore County, Baltimore.

Tyson, N. & Matula, V.C. (2004). Improved LSI-Based Natural Language Call Routing Using Speech Recognition Confidence Scores, *In Proceedings of EMNLP*, 2004.

Wild, F., Haley, D., & Bülow, K. (2011). Using latent-semantic analysis and network analysis for monitoring conceptual development. *Journal for Language Technology and Computational Linguistics*, 26(1), pp. 9–21.

Appendix:

Destinations:

Activate or disactivate a line, Check balance, Voicemail, Coverage, Contract details, Internet tariff, Businesses, Phonebill, Lost or stolen, General information, Internet, Logistics, Migration, Games and Software, Special offers, Call options, Agent, Switching service provider, Top-ups, Complaint, Roaming, Credit Balance, SMS and MMS, Call rates, TV, Phones, Shops, Languages, Incidents.