SWARM ALGORITHMS FOR NLP - THE CASE OF LIMITED TRAINING DATA

George Tambouratzis*, Marina Vassiliou

Institute of Language & Speech Processing,
RC "Athena" Artemidos 6 & Epidavrou str., 15125 Athens, Greece
*E-mail: giorg_t@ilsp.gr

Abstract

The present article describes a novel phrasing model which can be used for segmenting sentences of unconstrained text into syntactically-defined phrases. This model is based on the notion of attraction and repulsion forces between adjacent words. Each of these forces is weighed appropriately by system parameters, the values of which are optimised via particle swarm optimisation. This approach is designed to be language-independent and is tested here for different languages.

The phrasing model's performance is assessed per se, by calculating the segmentation accuracy against a golden segmentation. Operational testing also involves integrating the model to a phrase-based Machine Translation (MT) system and measuring the translation quality when the phrasing model is used to segment input text into phrases. Experiments show that the performance of this approach is comparable to other leading segmentation methods and that it exceeds that of baseline systems.

Keywords: particle swarm optimisation, natural language processing, text phrasing, machine translation

1 Introduction

The work presented here concerns the development of a phrasing model, which can be used for segmenting sentences of unconstrained text into syntactically-defined phrases (i.e. noun, verb, adjective phrases etc.), based on a limited amount of training data. The ultimate aim of creating this phrasing model has been its integration into a corpus-based MT system that by design can operate with only limited-scale bilingual resources.

Unsupervised parsing, which involves defining syntactic phrases, has been under study since the early 2000s in Natural Language Processing (NLP). Klein and Manning [1, 2] have proposed methods for inducing, in an unsupervised manner, hierarchical syntactic structure in arbitrary sentences us-

ing Expectation Maximisation (EM), these methods achieving class-leading accuracies. Raw text unsupervised parsing has also been studied by Seginer [3], using local predictions, and by Ponvert et al. [4], who employ Cascaded Finite State models.

Yarowsky and Ngai [5] and more recently Zhu et al. [6] have studied the porting of a phrasing model between two languages under the assumption that one of the languages has fewer resources. Thus its model can be created by modifying as appropriate a phrasing model established in the more amply-resourced language.

A distinct yet conceptually related research activity has involved segmenting continuous text into words for a number of Asian languages including Japanese and Chinese, as discussed in [7, 8, 9]. For

these languages, there exists the additional problem of splitting a sentence into words, since no spaces separate character sequences. As a result, the methods developed in these works require a substantial amount of training data, as the phrasing task includes the identification of lexical units. However, these methods are not particularly relevant to the task studied in the present article, which focuses on European languages where the boundaries between words are explicitly defined. Hence, here the phrasing algorithm only needs to determine phrase boundaries and not word boundaries as well.

2   Phrasing model outline

In the present article the Attraction-Repulsion phrase Generator model (henceforth also referred to as ARG) is evaluated in detail. ARG detects the attraction and repulsion forces that are established between adjacent words, in order to identify phrase boundaries.

The setup used is broadly similar to that of [6]. Our work uses a parallel bilingual corpus to extract a phrasing model, which is subsequently applied to the PRESEMT MT system (see [10] for a comprehensive description of PRESEMT). However, in contrast to Zhu et al. [6], who employ a parallel corpus of 10,000 sentence pairs in English and Chinese, the corpus used in the present work consists of only a few hundred sentences in the source and target languages. Such a small corpus suffices in our case, where the aim is not to generate syntactic graphs or to extract object/subject relations, but to perform a shallow parsing by defining (i) the phrase boundaries and (ii) the type of each phrase.

To represent the phrase segmentation, we adopt the IBO2 annotation scheme [11], which establishes phrase boundaries by assigning to the current word the symbol "B", if it corresponds to the start of a phrase, or the symbol "I", if it corresponds to the continuation of the phrase.

ARG is applied here to two European languages from different families, namely Greek and English, in order to gauge the wider applicability of the methodology across languages.

2.1   Relevant phrasing approaches

PRESEMT is a corpus-based MT system which extracts the language model from large monolingual corpora. It employs a phrase-based principle, where sentences are split into sub-sentential units which coincide with syntactic phrases.

When developing the PRESEMT methodology, multiple phrasing approaches have been tested, including approaches based on Conditional Random Fields (CRF) and Template Matching (TEM). CRF [12] is a structured learning algorithm, very frequently chosen for modelling tasks (cf. for instance [13, 14]). CRF intrinsically takes into account context to create a more accurate model. In the case of phrasing word sequences from texts, context takes the form of the adjacent phrase types and boundaries. When tested within PRESEMT, CRF provided a more accurate phrasing than other stochastic models such as Hidden Markov Models, yet it yielded suboptimal results. These results are most likely due to limited training data (circa 200 sentences, or roughly 2,000 phrases) that does not suffice to determine precisely the parameter values of the complex underlying model.

The TEM method [15] employs a greedy search to identify phrases that match patterns encountered during training. This search tries to match word sequences of the input sentence with a list of valid phrases extracted from the training dataset and sorted in descending order of expected accuracy. Experimental results [16] have shown that, for a limited-size training set, TEM is more effective than CRF in building a phrasing model. However, a key drawback of TEM is that if a suboptimal phrase is determined at some point, this decision cannot be reversed and so frequently leads to a build-up of errors. Furthermore, the use of an ordered list of phrases assumes that a human expert specifies appropriate ordering rules. These rules potentially vary not only for different languages but also amongst different domains of the same language.

2.2   The ARG model

The Attraction-Repulsion phrase Generator model is trained on a set of sentences, which are annotated with Part-of-Speech (PoS) tags and segmented into labelled phrases. When examin-

ing the tags of consecutive words in unseen sentences, ARG makes use of the training set to decide whether the given words belong to the same phrase, having stronger attraction forces, or whether they are separated by a phrase boundary (i.e. they are parts of different phrases) thus obeying the repulsion forces. For instance, a noun-verb sequence is split into two parts by a boundary between these two words, the first word belonging to a noun phrase whilst the second forming part of a verb phrase. By accumulating such information over the entire training set, ARG learns to identify phrase boundaries and segment unconstrained text into phrases, by tuning specific parameters termed weights.

Three variants of the Attraction-Repulsion model have been used to determine the most effective one (as detailed in Section 3.3). This is motivated by the fact that the amount of training data is limited; hence data may not be sufficient to train a model with a relatively large number of parameters. Besides, it is preferable to have a more constrained set of parameters that take independent values. This is also the idea underlying idea of Booker et al. [17], who proposed the use of surrogate objective functions with a reduced number of parameters to guide an optimisation process, in order to achieve a better optimisation point. The use of surrogate functions to improve the effectiveness of swarm optimisation methods has been reviewed by Jin [18].

2.3   Identification of phrases

Segmenting a sentence with n words into phrases means determining whether at the point just after the ith word a boundary is needed or not. This is decided based on the identities of the words at the vicinity of this point. The phrasing task is schematically depicted in Figure 1.

One could envisage this as a situation where neighbouring words are attracted to or repelled from each other. Attractive forces indicate that the corresponding words are closely related and belong to the same group, i.e. the same phrase, in the given context. Strong repulsive forces indicate these words are at a large distance from one another and should be assigned to different phrases. This modelling mimics that of natural systems, where multiple forces come into play between elements to reach an equilibrium. For each candidate segmentation point, a set of attraction and repulsion forces

are identified over different distances, taking into account varying contexts to determine a constituent force.

The concept of attraction-repulsion paired forces has been adopted in several machine learning problems. In the context of this article, the closest attraction-repulsion application is that of Stoeber et al. [19] for phrasing purposes with a Text-to-Speech synthesis system. However, our set-up differs as the amount of training data is much smaller. Furthermore, the phrasing scheme is designed with a view to being integrated to an MT system which employs only limited training data.

3   Defining the attraction and repulsion forces

An attractive or a repulsive force (represented as *Fatt(X)* and *Frep(X)* respectively, where X denotes the candidate segmentation point) is always defined over a specific range of observations (in this case words). A number of ranges are used in the system to combine both information that is localised (for instance only one or two words directly adjacent to the candidate segmentation points) as well as more global, involving more distant words. This range is defined as a set of k datapoints

$$range_j = [W_{j,1}, W_{j,2}, .., W_{j,k}]. \quad (1)$$

To determine the value of the jth force (either attractive or repulsive), a ranking function $?$ is defined which reflects the importance of a consecutive set of words (this could be the frequency of occurrence of the word sequence). For instance, the definition of attractive force $Fatt_j(X)$ takes into account word sequences from the training set where there is no segmentation boundary. On the other hand, the repulsive force $Frep_j(X)$ takes into account training patterns where there is a phrase boundary at position X. For example, an attractive force between two consecutive words is expressed as follows

$$Fatt_j(X) = \varphi(W_{j,x} = word_x; W_{j,x+1} = word_{x+1},$$
with no phrase boundary between words x & x+1,
$$\quad (2)$$

where $\varphi()$ is calculated over the entire set of positive training patterns (as discussed in Section 3.2).

Input sentence: "The child runs to school."

Sentence segmented into phrases: "(The child) (runs) (to school)"

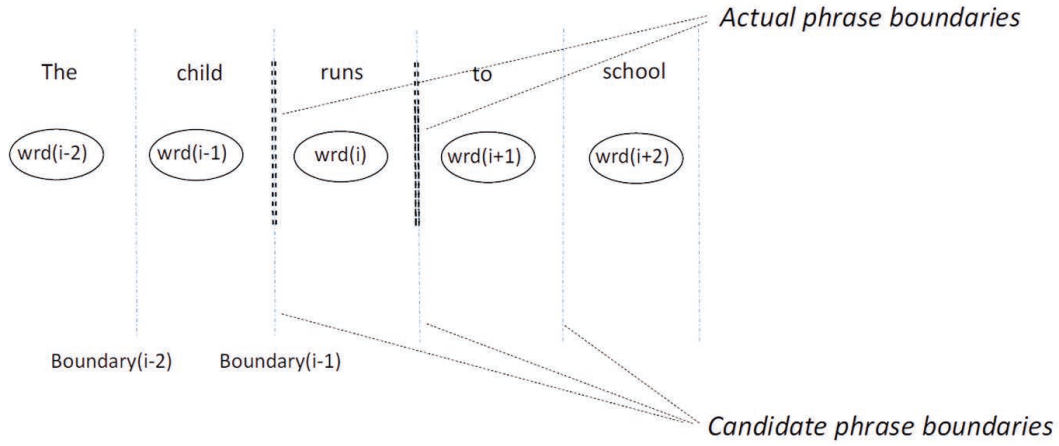Sentence expressed in part-of-speech tags: "(At No) (Vb) (Prp No)"

Figure 1. Example of the phrasing task

An analogous expression is used to define repulsive forces. This likelihood of a segmentation point is determined by combining the evidence within the training set of the attractive and repulsive forces at the specific point. These forces combine to a single potential value, denoted as *Pot*, which is defined as

$$Pot\,(X) = \prod_{i=1}^{k} a_i \cdot Fatt_i\,(X) - \prod_{i=1}^{k} b_i \cdot Frep_i\,(X) - thres.$$
(3)

In the present work, a total of eight constituent forces have been defined, sampling from 1 to 4 words. These are organised into four pairs of attractive and repulsive forces (and thus $k=4$), as shown in Figure 2. In each pair, the attractive and repulsive forces sample the same range (set of words), but split over two different training sub-sets. Attractive forces take into account training patterns where there is no phrase boundary for the given point. On the contrary repulsive forces only use training patterns for which a phrase boundary exists.

As expressed in (eq.3), the attractive and repulsive forces are multiplied by weights a_i and b_i, respectively. The magnitudes of these weights indicate their relative importance in establishing whether a boundary should be placed between two words or not. For instance, the frequency of a pat-

tern with a range of 2 (i.e. comprising two words only) is expected to be much more frequent in the training set than a pattern with a range of 4 words. Yet, the information provided by the range-4 pattern is probably much higher and thus a larger weight must be used. Finally *thres* is a bias that establishes the minimum level of the potential (*Pot),* below which a decision is made to introduce a phrase boundary.

A positive value of the potential indicates that at point *i* there is no phrase boundary as the attractive forces exceed repulsive ones, while a negative value indicates the need for a phrase boundary. The magnitude of the potential indicates the confidence of the decision of placing or not a boundary. In order to segment a set of sentences, (eq.3) is applied independently to every possible segmentation point between consecutive words, to establish the actual phrase boundaries.

3.1   Determining function φ

To calculate the attraction and repulsion forces via (eq.2) and (eq.3) there is a need to define **ranking function φ**. The ranking function expresses the relative importance of sub-sentential segments in the training corpus, allowing them to be ordered based on their importance. In general, function φ
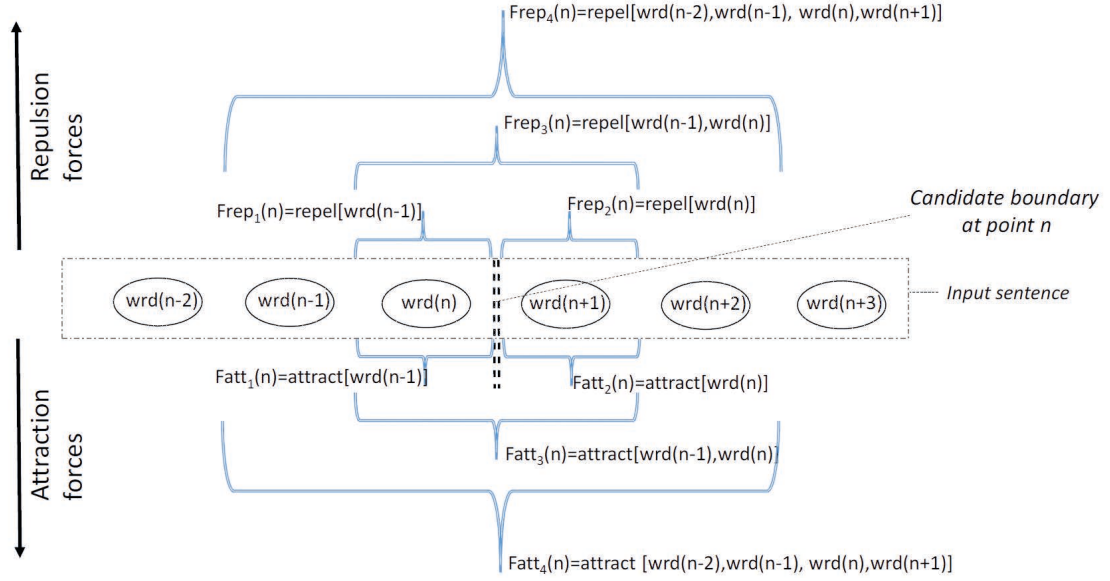
Figure 2. Ranges over which attractive & repulsive forces are calculated for the phrasing task

should emphasise the most frequent patterns, reflecting their higher frequency of occurrence. In the present article, two functions are used, φ_1 which is based on frequencies solely and φ_2 which combines frequency information with the length of the phrase in tokens. Function φ_2 compensates for the expectedly lower frequencies when a sub-sentential phrase consists of more tokens. The two functions are defined as follows

$$\varphi_1(phr) = freq(phr), \qquad (4)$$

$$\varphi_2(phr) = freq(phr) + r \cdot len(phr), \qquad (5)$$

where $freq(phr)$ represents the frequency of occurrence of a phrase pattern in the training set and $len(phr)$ is the length in tokens of the phrase pattern phr. In (eq.5), r is a weighing factor which in the experiments reported hereafter is set to 100.

3.2 Extracting information from the training set

In the ARG implementation, PoS tags are employed instead of words to address the data sparsity issue, since frequencies are counted over a more limited vocabulary of features. The use of PoS tags enhances the ability of the system to learn over limited-size training sets, in comparison to the use of words.

ARG phrasing distinguishes two tasks, (1) definition of phrase boundaries and (2) labelling, i.e. identification of phrase types. To implement the first task, a list of all boundary phrase templates (BNDC), i.e. all sequences of tags from the training set with a phrase boundary at the specific point, is established. Similarly, a list is established of all in-phrase templates (INPC), where there is no such boundary. Templates of type BNDC and INPC provide the training data for repulsive and attractive forces (*Frep(i)* and *Fatt(i))* respectively. In conceptual terms, BNDCs form the negative training set as they strengthen repulsive forces, while INPCs form the positive training set as they support attractive forces.

The second task involves identifying the phrase type. This is achieved by incorporating all phrase templates from the training set in a labelled list, and ordering them on their frequency. Then a new phrase (determined as the sequence of tags between two consecutive phrase boundary points) is assigned the type of the most frequent entry from the labelled list that comprises the exact same sequence of tags.

Table 1. Templates extracted for the sentence of Figure 1, expressed in terms of words and PoS tags

id.	tokens	PoS tags	type	Freq.
1	(the, child)	(At, No)	INPC	1
2	(child, runs)	(No, Vb)	BNDC	1
3	(runs, to)	(Vb, Prp)	BNDC	1
4	(to, school)	(Prp, No)	INPC	1

The steps of establishing an ARG phrasing model are listed in Figure 3. As an example, Table 1 contains templates extracted from the short sample sentence of Figure 1. For clarity, only the in-phrase and boundary templates that can be extracted for two consecutive words (one to the left and one to the right of the candidate segmentation point) are shown in Table 1. For the sample sentence, there are four possible segmentation points, and thus four templates are extracted. The first and fourth are IN-PCs (as there is no segmentation boundary between the words), whilst the second and third are BNDC templates. Obviously, as more sentences are added to the training set, the coverage of the training set increases substantially and BNDC and INPC patterns get a differing frequency counts.

For a typical dataset as used in the present work, these templates are extracted by processing a training set of 200 sentences. Using the process of Figure 3 for the experimental corpus the total number of distinct boundary templates and in-phrase templates is 1,322 and 1,036 respectively (more information on the datasets is provided in Section 4).

A. Training
Step 1: For the sentences of the training corpus, replace tokens with PoS tags.
Step 2: Compile set of phrase templates from training corpus.
Step 3: Compile set of boundary templates from training corpus.
Step 4: Compile set of in-phrase templates from training corpus.
Step 5: For each template range record attraction and repulsion templates and count frequencies over the training corpus
Step 6: Based on the frequencies of step 5 calculate frequency-related metric of performance by applying definition of φ
Step 7: Determine attraction/repulsion weights algorithmically to optimize performance.

B. Application
Step 8: Calculate potential for each possible boundary position of sentence.
Step 9: Establish boundary points by comparing with threshold value
Step 10: Determine phrase label for each sequence within boundaries by comparing to phrase templates

Figure 3. Steps for training and applying the attraction-repulsion phrasing model

3.3   Establishing the optimal ARG parameter values

From the training patterns, ARG needs to determine the optimal weights for appropriately combining the constituent forces, to decide if a phrase boundary is needed or not at a specific point. For the implementation studied here, there are nine weights in total (cf. eq.3), whose values must be determined. To that end, we utilise the Particle Swarm Optimisation (PSO) method from the field of computational intelligence. PSO [20] implements a search for the optimal solution by replicating the collective behaviour of a swarm of living organisms, when for instance these organisms search for food. The PSO principle is to use a set of simple agents (also called particles) whose searches are appropriately combined, via exchange of information over a series of training epochs.

Each particle is determined by its current location in the pattern space, $x_i(t)$ (corresponding to a candidate solution), and its current velocity vector $v_i(t)$ (the speed as well as direction with which the agent searches through the pattern space for new solutions). As the swarm evolves, the location and velocity of the ith particle are updated in accordance to the best solution $P_i(t)$ found by this particle and the global-best solution $P_g(t)$ found by all particles

$$v_i(t+1) = w \cdot v_i(t) + c \cdot U[0,1] \cdot (P_i(t) - X_i(t))$$
$$d \cdot U[0,1] \cdot (P_g(t) - X_i(t)) \quad , \tag{6}$$

$$x_i(t+1) = x_i(t+1) + v_i(t+1). \tag{7}$$

Parameter w is termed the inertia weight of the swarm. Parameters c and d are known as the cognitive and social acceleration coefficients, and control the amount of local and global search respectively. $U[0,1]$ represents a random number drawn from a uniform distribution within the range from 0 to 1.

A basic PSO algorithm has been applied to the task of ARG weight selection [21]. In the present article, results on the latest experiments are presented, comparing two PSO variants. The first variant is sPSO (standard PSO). sPSO closely follows the initial model of Kennedy and Eberhart [20], augmenting it with mechanisms for (a) elitism (the best solution in the swarm is carried unchanged to

the next epoch), (b) velocity reinitialisation (which reduces premature convergence by increasing the magnitude of velocity if it falls below a threshold) and (c) fine-tuning after each swarm epoch, to implement local search. In sPSO, a fully informed swarm is used, where all particles are aware of the best solution found at each epoch [22].

The second variant considered is Adaptive PSO (AdPSO), proposed by Zhan et al. [23]. In AdPSO, a more elaborate search is employed, with parameters c and d of (eq.6) being algorithmically varied without any human intervention during run-time. This enables the swarm to switch between different search strategies as it evolves, to optimise its performance. AdPSO has been found to be more effective than sPSO in a collection of standard test functions [23].

For both AdPSO and sPSO, 1,000 epochs were simulated per run, with each experiment repeated for a total of 100 randomly initialised runs. Each swarm comprises a total of 20 particles. Regarding the potential defined in equation (3), three different **weight configurations** are defined, depending on whether the weight parameters are independently optimised or not. These are:

1. **exact**, where a_i and b_i take independent values throughout all epochs,

2. **surrogate**, where for all epochs $a_i = b_i$,

3. **hybrid**, where for a number of initial learning epochs (typically 20% of the available epochs) weights a_i and b_i share the same values for every i, and for the remaining epochs they are decoupled to evolve independently.

Hence, the use of two swarm variants (sPSO and AdPSO) and three weight configurations (exact, surrogate and hybrid) gives a total of six different configurations to run and compare for each experiment set.

4 Experimental set-up & results

The evaluation of the ARG-based model is twofold; on the one hand, the phrasing accuracy is assessed by comparison to a reference segmentation drawn up manually. On the other hand, the effect of the phrasing model on the translation output of PRESEMT is assessed by comparing the translation quality to the quality achieved by the use of other phrasing models.

In our experiments, three text corpora are used, each comprising 200 sentences. The first corpus is used for training, and the second set is used for development purposes. The first and second corpora are similar, with differences limited to about 10% of the word tags. The third corpus, which also comprises 200 sentences, is completely unrelated to the training and development sets, and serves to test the produced phrasing model with a fully independent set of unseen sentences.

The experiments presented here represent a substantial expansion in comparison to the experiments reported in [24]. More specifically, the swarm algorithms have been re-run, after incorporating numerous refinements and enhancements. These include the random initialisation of particle parameters, now using values that are much closer to the middle of the range, to have a less biased initial position and a more modest starting velocity of particles in the pattern space. In addition, the velocity update rule has been refined to perform a more gradual convergence of particles to their final values. To this end the velocity vector is normalised to a lower level throughout the experiments (0.125), for a finer convergence process.

Finally, an extensive statistical analysis of the solutions returned by the swarm algorithms is performed. The aim is to identify the best choice of swarm variant and weight configuration.

4.1 Phrasing accuracy of ARG

Experiments were carried out to optimise the ARG weight values, using both sPSO and AdPSO. To evaluate the quality of the phrasing solution generated by each swarm particle during evolution, a metric is required. Experimentation has shown that using metrics such as recall, the phrasing model is led to add more segmentation points. With recall, an optimal score would be achieved even in the trivial case where all possible segmentation points become phrase boundaries, since recall does not penalise the introduction of erroneous phrase boundaries. As a consequence the actual phrasing accuracy suffers. Thus, it is preferable to use metrics that take into account both where boundaries should be placed and

where they should not, and penalise both types of errors. For this reason, the F-measure is used, as expressed in equation (8).

$$F - measure = \frac{tp}{tp+tn} \cdot \frac{tp}{tp+fn}. \qquad (8)$$

The results obtained by different ARG configurations are depicted in Tables 2 and 3, in terms of phrasing accuracy, by comparison to a golden segmentation manually compiled. In both Tables 2 and 3, the best weight configuration per PSO variant and ranking function combination is marked by an asterisk. The best result for sPSO and AdPSO is denoted in bold, whilst the best result over all combinations is denoted by a highlighted entry.

Since a relatively large number of experiments have been run, the effect each factor has on the outcome of the swarm optimisation is presented in a separate sub-section hereafter.

Comparing swarm algorithms AdPSO and sPSO: Concentrating initially on the results for the English language (Table 2), the first observation is that, in general, AdPSO is more effective than sPSO in minimising the phrasing error. This holds for both φ_1 and φ_2. Similar observations apply when training a phrasing model for the Greek language based on the results of Table 3.

Table 4. Comparison of swarm algorithms in terms of the average phrasing error (over 100 runs) when using function φ_2

	Pot **Funct**	**sPSO error**	**AdPSO error**	**error reduct.** *(AdPSO over sPSO)*
EN	Exact	0.0679	0.0565	-16.8%
	Surrogate	0.0668	0.0538	-19.5%
	Hybrid	0.0660	0.0532	-19.4%
GR	Exact	0.0861	0.0308	-64.2%
	Surrogate	0.0746	0.0299	-59.9%
	Hybrid	0.0760	0.0298	-60.8%

It is interesting to quantify how the two swarm algorithms compare to each other in terms of the solution achieved. The improvements of AdPSO over sPSO are presented in Table 4 for the average error, calculated over the 100 randomly-initialised runs. A comparison of the lowest error over 100 runs is summarised in Table 5, for the particular case of function φ_2 (similar observations apply to function φ_1). Both tables confirm that AdPSO consistently

achieves a substantially lower error than sPSO (and thus a higher phrasing accuracy). The reduction in error varies from 15% to more than 60%, reflecting the superior performance of AdPSO throughout the reported experiments.

Table 5. Comparison of swarm algorithms in terms of the minimum phrasing error (over 100 runs) when using function φ_2

	Pot **Funct**	**sPSO error**	**AdPSO error**	**error reduct.** *(AdPSO over sPSO)*
EN	Exact	0.0617	0.0519	-15.9%
	Surrogate	0.0589	0.0522	-11.4%
	Hybrid	0.0567	0.0510	-10.1%
GR	Exact	0.0599	0.0300	-49.9%
	Surrogate	0.0573	0.0297	-48.2%
	Hybrid	0.0614	0.0283	-53.9%

This indicates that the phrasing model created by AdPSO is consistently more accurate. In addition, according to Tables 2 (for the English language) and 3 (for the Greek language), AdPSO uses up a larger fraction of the available epochs before it settles to its final solution (which is of a higher quality). This illustrates the higher exploration ability of AdPSO in comparison to sPSO. Increased exploration is attributable to the more elaborate AdPSO algorithm, which allows it to alter the cognitive and social acceleration coefficients (parameters c and d of (eq.6)) so as to switch quickly between exploration and exploitation strategies and cover the search space to a greater extent.

Comparison of weight configurations: Another observation is that a better optimisation (i.e. a lower error rate) is achieved by a swarm algorithm employing a simplified objective function with fewer parameters. More specifically, for the sPSO variant, the surrogate and the hybrid configurations are more successful in determining a lower error than the exact configuration. The same observation also applies to AdPSO. A lower phrasing error is consistently obtained when the search space is reduced, by using a smaller set of independent parameters, whose values are estimated from the training set.

Regarding the minimum error achieved for the English language, according to the two right-most columns of Table 2, sPSO using the exact configuration settles more quickly than with the surrogate configuration. The hybrid configuration requires the largest number of epochs to settle.

Table 2. Results for sPSO and AdPSO over 100 randomly-initialised runs, for (i) ranking function φ_1 and (ii) ranking function φ_2, when segmenting English text into phrases

	Funct.	Crit	Error				Settling step (for run with lowest error)			
			min	max	median	st.dev.	min	max	median	st.dev
sPSO	Exact	φ_1	0.0851	0.1123	0.1007	0.0059*	14*	999*	296	302.9
	Surrog	φ_1	0.0850	0.1060*	0.0901*	0.0060	2	999*	483	292.9
	Hybrid	φ_1	0.0820*	0.1158	0.0963	0.0085	8	999*	486*	265.4*
	Exact	φ_2	0.0617	0.0775	0.0674	0.0038	**19***	**1000***	123	258.4
	Surrog	φ_2	0.0589	**0.0719***	0.0683	**0.0036***	11	999	125	307.2
	Hybrid	φ_2	**0.0567***	0.0749	**0.0666***	0.0038	17	999	**291.5***	**171.1***
AdPSO	Exact	φ_1	0.0758	0.0982	0.0860	0.0071	31	991	639.5	247.6*
	Surrog	φ_1	0.0749	0.0819	0.0750*	0.0032	24	996	161	2935
	Hybrid	φ_1	0.0720*	0.0788*	0.0773	0.0017*	188*	**998***	788*	2374
	Exact	φ_2	0.0519	0.0613	0.0558	0.0027	147	998	679	251.5
	Surrog	φ_2	0.0522	0.0590	0.0542	0.0010	45	998	473	330.0
	Hybrid	φ_2	**0.0510***	**0.0550***	**0.0534***	**0.0009***	**292***	989*	**809***	**214.1***

Table 3. Results for sPSO and AdPSO over 100 randomly-initialised runs, for (i) ranking function φ_1 and (ii) function φ_2, when segmenting Greek text into phrases

	Funct.	Crit	Error				Settling step (for run with lowest error)			
			min	max	Median	st.dev.	min	max	median	st.dev
sPSO	Exact	φ_1	0.1022	0.1226*	0.1187	**0.0033***	12*	**1000***	97	201.3*
	Surrog	φ_1	0.0994	0.1227	0.1070*	0.0060	7	**1000***	286.5	305.5
	Hybrid	φ_1	0.0970*	0.1230	0.1192	0.0069	0	999	299*	239.2
	Exact	φ_2	**0.0573***	**0.0794***	**0.0761***	0.0041*	10	999*	99	267.7
	Surrog	φ_2	0.0614	0.0807	0.0776	0.0041*	3	725	226.5	**136.8***
	Hybrid	φ_2	0.0596	0.0794	0.0769	0.0047	23*	978	**475.5***	269.6
AdPSO	Exact	φ_1	0.0721	0.1033	0.0839	0.0065	84	998	679	267.2
	Surrog	φ_1	0.0639	0.0719	0.0639*	0.0010*	187	994	872	279.4
	Hybrid	φ_1	0.0637*	0.1018	0.0673	0.0067	**244***	**999***	**947***	**179.3***
	Exact	φ_2	0.0300	0.0349	0.0306	**0.0006***	91	988	571*	251.9
	Surrog	φ_2	0.0297	0.0352	**0.0298***	0.0007	17	991	195	344.3
	Hybrid	φ_2	**0.0283***	**0.0342***	0.0300	0.0009	200*	996*	743	197.7*

The situation changes in the case of AdPSO, where the exact configuration requires a larger amount of epochs than the surrogate configuration before settling to a final solution. This again reflects the more elaborate algorithm of AdPSO which allows switching from exploration to exploitation and vice versa. In all cases reported in Table 2, when a hybrid configuration is used, the number of epochs is higher, though this is - at least partly - attributable to the need to complete all epochs with the surrogate function before optimisation with the exact function is initiated. Thus the swarm requires a minimum number of epochs before settling to a solution.

These observations also apply without significant differences when phrasing texts in the Greek language (cf. Table 3). With sPSO, a faster convergence is attainable using an exact function, whilst the use of surrogate functions necessitates more epochs. When using AdPSO with the exact configuration, the number of epochs required for the system to settle increases. In all cases, it is the hybrid configuration that requires the most epochs to settle to a solution.

Comparing ordering functions φ_1 and φ_2: Turning to the best choice of function φ for ordering the in-phrase and boundary patterns it can be seen that function φ_1 is not as successful in minimising the error as φ_2. More specifically, for phrasing English texts (cf. Table 2), for all three configurations of the parameter set (exact, surrogate and hybrid), sPSO achieves a lower error rate (i.e. fewer erroneous phrase boundaries) when using φ_2 rather than φ_1. The same observations apply to AdPSO, as can be seen in the second half (lower 6 rows) of Table 2 for the English language. This is attributable to the use of information concerning phrase length in addition to frequency in function φ_2, creating a more accurate grading. This leads to fewer phrasing errors for φ_2, as compared to φ_1.

Once again, the same observations apply to the phrasing of Greek texts, based on the contents of Table 3.

4.2 Visualisation of Solutions

An intuitive visualisation of the aforementioned error populations is provided in the form of boxplots. Boxplots depict the distribution of final phrasing errors for every set of 100 runs. One boxplot is provided for each of the swarm variants (sPSO or AdPSO) when one of the three weight configurations is used (either exact, surrogate or hybrid). Thus, in each figure, six boxplots are included, the three leftmost configurations corresponding to sPSO and the three rightmost ones to AdPSO.

The boxplots for the English language using φ_1 are shown in Figure 4. In this case, the errors generated by AdPSO (around 6.5%) are consistently lower than those of sPSO (which range from 9 to 12%). In addition, the use of surrogate and hybrid configurations generally results in lower errors in comparison to the exact configuration for both sPSO and AdPSO. For sPSO, the lowest average error occurs for the surrogate function, while in the case of AdPSO the lowest average error is achieved with the hybrid configuration. In addition, the average errors obtained with AdPSO for the exact, surrogate and hybrid configurations are very close to each other. This contrasts the behaviour of sPSO, where differences in the average errors are more substantial.
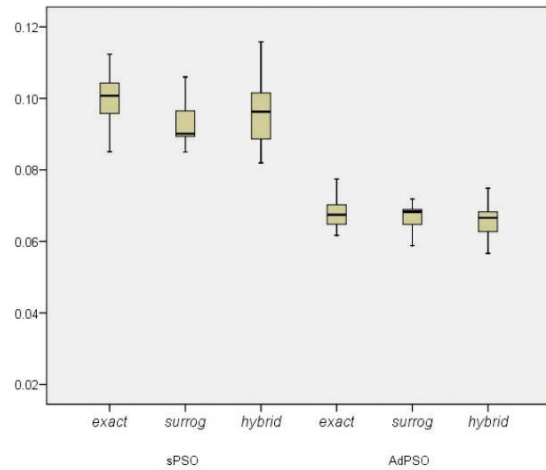

Figure 4. Population of phrasing errors for English (over 100 runs) for each combination of swarm variant and weight configuration, using function φ_1

Regarding the best solutions generated by each configuration, (see Figure 4), the difference is more marked. The exact configuration generates a higher error than the surrogate one, which in turn generates a higher error than the hybrid configuration. This observation applies to both sPSO and AdPSO.
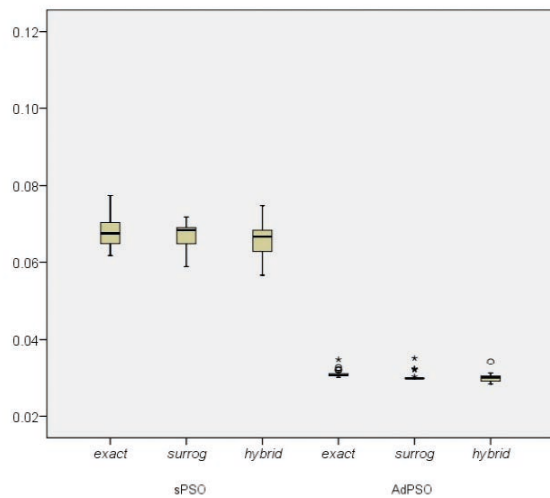
Figure 5. Population of phrasing errors for English (over 100 runs) for each combination of swarm variant and weight configuration, using function φ_2.

In Figure 5, the corresponding boxplots are shown for English texts when ordering function φ_2 is used. Similarly to when using φ_1 (cf. Figure 4), the solutions generated by AdPSO with φ_2 are superior to those of sPSO. The use of ordering function φ_2 instead of φ_1 results in less variability in the error for all three weight configurations. Also the magnitude of the error is substantially reduced, from approx. 10% for sPSO with φ_1 to 6.5% for φ_2. For AdPSO, the results are respectively 6.5% with φ_1 as compared to 3.5% with φ_2.

When creating a phrasing model for Greek, the corresponding error populations are shown in Figures 6 and 7, for ordering functions φ_1 and φ_2 respectively. The results for Greek are in agreement with the aforementioned results for English, indicating a consistent performance of ARG across languages: AdPSO produces a better phrasing in comparison to sPSO, whilst φ_1 generates a higher error than φ_2.

When using φ_1, the surrogate and hybrid configurations achieve lower mean errors than the exact configuration, both for AdPSO and sPSO. A minor variation occurs for function φ_1, for Greek, where the surrogate configuration leads to the lowest error, which is lower than the hybrid configuration. The situation changes when ordering function φ_2 is used, as the errors are substantially reduced for all configurations and now the hybrid configuration results in a slightly lower error rate than the surrogate

one.

As a conclusion, the experimental results are found to be consistent across languages. Apart from involving fewer computations and thus being more efficient to simulate (as the number of weights is substantially smaller), surrogate and hybrid configurations have the benefit of reaching a lower error than the exact configuration. Thus, the strategy of defining groups of weights that take the same values appears to be effective, yielding a phrasing model which more accurately replicates the desired phrasing.
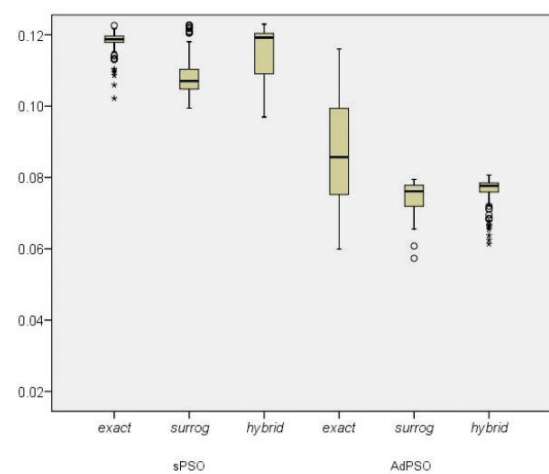

Figure 6. Population of phrasing errors for Greek (over 100 runs) for each combination of swarm variant and weight configuration, when using function φ_1
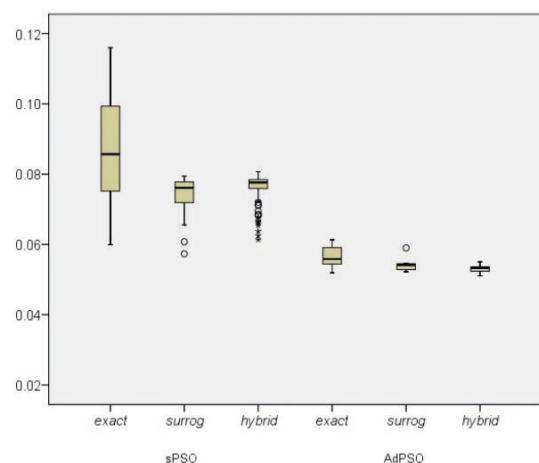

Figure 7. Population of phrasing errors for Greek (over 100 runs) for each combination of swarm variant and weight configuration, when using function φ_2.

4.3 Translation quality using ARG

As noted before, the effectiveness of the ARG phrasing model is additionally assessed in the machine translation task. In the experiments reported here, the PRESEMT phrase-based MT system is used to translate text segmented into phrases by the ARG model.

The quality of the translation is measured via two widely-used objective metrics, BLEU [25] and NIST [26], which produce a score by comparing the system translation to a set of reference translations. For both metrics, a higher score indicates a better translation. The values of BLEU range from 0 up to 1, with 1 indicating a perfect translation, whilst the values of NIST range from 0 to infinity.

For this evaluation the language pairs used were from Greek to English and vice versa. The Greek phrasing model was used to segment Greek text being translated to English, and feed this to the PRESEMT phrase-based system, whilst the English phrasing model was used in the opposite translation direction. The results obtained are depicted in Tables 6 and 7, for the Greek and English ARG phrasing models respectively.

Comparison of swarm variants: According to Table 6, the BLEU scores do not reflect clearly the lower phrasing error of AdPSO, as the best BLEU scores are achieved for sPSO rather than AdPSO. An analysis of the translation results has shown that this is mainly due to the fact that AdPSO-derived modelling forms longer phrases (including more words between consecutive phrase boundaries) in comparison to sPSO. However, the target language model used in PRESEMT provides a better match to the shorter phrases generated by sPSO rather than the longer phrases of AdPSO (though these latter phrases are still grammatically correct).

Table 6. Best results over 100 randomly-initialised runs when optimising phrasing with φ_1 for **Greek** (used to translate from Greek to English)

	Pot Funct	error	Settling epoch	BLEU	NIST
sPSO	Exact	0.1022	570	**0.382***	**7.300***
	Surrog	0.0994	881*	0.360	7.096
	Hybrid	0.0970*	397	0.375	7.201
AdPSO	Exact	0.0721	695	0.368	7.165*
	Surrog	0.0639	751	0.368	7.136
	Hybrid	**0.0637***	**868***	0.373*	7.159

When using the ARG model to segment English text into phrases (see Table 7), the best NIST score, indicating a better translation performance, is achieved for AdPSO. What is also notable is that for English, all configurations give very similar phrasing error rates.

Table 7. Best results over 100 randomly-initialised runs when optimising phrasing with φ_1 for **English** (used to translate from English to Greek)

	Pot Funct	error	Settling epoch	BLEU	NIST
sPSO	Exact	0.0851	908	0.126	**4.011***
	Surrog	0.0850	499	0.124	3.978
	Hybrid	0.0820*	794	**0.130***	4.005
AdPSO	Exact	0.0758	872	0.126*	3.993*
	Surrog	0.0749	720	0.125	3.972
	Hybrid	**0.0720***	240	0.125	3.972

Swarm optimisation with full parameter sets vs. reduced parameter sets: Regarding the configuration used (the number of independent weights), for English the exact configuration and the hybrid configuration result in higher BLEU scores (cf. Table 7), whilst the surrogate configuration results in an inferior translation result.

By referring to Table 6, it can be confirmed that these general observations for English also apply when phrasing texts in the Greek language. This indicates the consistently successful application of PSO in optimising the parameters of the ARG phrasing model, irrespective of the language being used.

It is also of particular interest to determine how ARG compares to other phrase-generating methods when integrated to the PRESEMT translation system. This is discussed in the next sub-section.

4.4 Comparing the ARG phrasing to other models

Within the second evaluation method, the ARG model is compared to other phrasing models in terms of the final translation accuracy. For comparison, we have chosen the phrasing models of CRF and TEM, both of which have been integrated into the PRESEMT translation system in the past. The evaluation results obtained are presented in Table 8.

Table 8. Metric scores obtained for the three phrasing methods, the best score per metric denoted by an asterisk

	Greek phrasing model		English phrasing model	
	BLEU	NIST	BLEU	NIST
CRF	0.342	7.067	0.120	3.770
TEM	**0.393***	**7.345***	0.118	3.862
ARG/sPSO	0.382	7.300	**0.130***	**4.005***
ARG/AdPSO	0.373	7.160	0.126	3.993

The highest performing CRF variant is trained by using 1-grams and 2-grams which involve the current and previous words within a sentence environment (each word is represented by its PoS tag). This model has given the optimal results for the PRESEMT training sets.

Even though CRF is widely accepted in literature as the default choice for phrasing, the ARG results for both English and Greek and both swarm variants, as shown in Table 8, are measurably better than those of CRF. This indicates that the attraction-repulsion concept yields an accurate phrasing model. In addition, ARG clearly provides the rationale of a phrasing decision as both the training data and the weights are easy to inspect. This is in contrast to the mathematically complex CRF model, where the final values of parameters are not transparent.

Based on the experimental results, ARG is superior to TEM when creating a phrasing model for English, whilst TEM is in turn superior when segmenting Greek text into phrases. The comparatively lower ARG phrasing quality for the Greek language is mainly due to frequently-occurring issues, which are traceable to the existing PRESEMT language models:

1. The inability to correctly segment complex noun phrases, which are relatively infrequent in the training data but occur more frequently in the test data.

2. In contrast to English, Greek is a highly-inflectional language with four different grammatical cases. Since case information is included in the PoS tag of nominal elements (e.g. nouns, adjectives, participles etc.), the phrasing task is compounded by a larger number of possible tags, the amount of training patterns per tag being substantially reduced for less frequent

case patterns. Thus, the system is less accurate in generalising from known patterns to other unseen (within the training corpus) patterns (such as different cases).

3. A third issue emerges as a tendency to position verbs and article-less direct objects into the same phrase (e.g. κάνω διακρσεις = *lit. make discriminations*, discriminate). This is due to language-specific structures, frequently-appearing in the training data, involving a verb plus a direct object pattern in the source language corresponding to a single verb in the target language.

5   Conclusions & future directions

In the present article, a method based on attraction and repulsion forces has been proposed to identify syntactic phrases for arbitrary text. This method is designed specifically for extracting knowledge from a relatively small number of sentences. This allows the method to be easily trainable for a variety of languages, even when language resources are scarce. In this article, results have been reported for both Greek and English, to comprehensively examine the suitability of the proposed method to texts written in different languages.

Concerning variants of swarm optimisation, the error rates achieved by AdPSO are consistently lower than those achieved by sPSO. This illustrates a superior performance by the more sophisticated AdPSO algorithm in optimisation tasks, which allows it to switch more effectively between exploration and exploitation.

A series of experiments have focused on defining a subset of weights that are grouped together, so that optimisation is done in two phases. In the first phase only a smaller set of parameters are independently optimised, whilst in the second phase all parameters of the attraction-repulsion model are independently optimised (this being the hybrid configuration).

For both languages tested, it was found that the hybrid approach resulted in a lower error for the final solution (and thus a superior solution). This result is potentially of interest concerning the application of computational intelligence techniques to natural language processing tasks. The coupling

of specific parameters to share the same value allows the complexity of the optimisation task to be reduced via a divide-and-conquer approach, whilst after a good solution has been found all parameter values can be independently optimised.

Experiments show that the attraction-repulsion method, with weights optimised via PSO, generates a phrasing model superior to the CRF model (the long-established standard method for phrasing models) in terms of both the phrasing accuracy as well as the translation accuracy when the phrasing model drives a phrase-based MT system. Notably, the performance of ARG in the phrasing task is also very close to that of the TEM method which has been manually fine-tuned. This observation illustrates the effectiveness of the swarm optimisation in determining optimal weights.

An added benefit of using the attraction-repulsion concept is that parameters are set by an automated optimisation process, and can be directly interpreted and even modified. Other metaheuristics besides PSO are of course valid alternatives for optimising the attraction/repulsion weights but are the topic of future work.

An additional aspect for future study is that of using more detailed tag information in order to improve the phrasing accuracy. For instance, by differentiating between auxiliary verbs and main verbs in the English language, an improved phrasing performance is achieved, which also translates to a higher translation quality. This opens the field for further improvements in Machine Translation quality.

However, the main projected improvements focus on the swarm algorithm details. One key observation is that in terms of phrasing error sPSO has been consistently outperformed by AdPSO, due to the latter model's ability to switch between exploitation of already determined solutions and exploration to determine new solutions. The aim is to utilise recent research findings to investigate if sPSO performance may be improved. In this aspect the work of Harrison et al. [27] on systematically establishing the most relevant values for key parameters such as the inertia, cognitive and social coefficients is particularly relevant. This work will show if sPSO can become competitive to AdPSO in terms of phrasing accuracy.

Another finding of the present article is that AdPSO frequently separates sentences into fewer phrases than sPSO in certain configurations. This is due to the fact that the F-measure metric is not symmetric in terms of the establishment of phrases but favours avoiding to establish incorrect phrases. Because of this, it is planned to evaluate other metrics such as Youden's metric and Cohen's K in the near future which are more balanced in considering the accuracy of both the positive and negative instances (in the present context, the existence and the absence of phrase boundaries). Also it is intended to analyse in detail the swarm optimisation process to determine if certain aspects can be further improved.

6   Acknowledgements

References

[1] D. Klein and C. D. Manning, A generative constituent-context model for improved grammar induction, Proceedings of 40th ACL Meeting, Philadelphia, USA, pages 128–135, July 2002.

[2] D. Klein and C. D. Manning, Corpus-based induction of syntactic structure: Models of dependency and constituency, Proceedings of 42nd ACL Meeting, Barcelona, Spain, pages 478–485, July 21-26, 2004.

[3] Y. Seginer, Fast unsupervised incremental parsing, Proceedings of 45th ACL Meeting, Prague, Czech Republic, pages 384–391, June 2007.

[4] E. Ponvert, J. Baldridge, and K. Erk, Simple unsupervised grammar induction from raw text with cascaded finite state models, Proceedings of 49th ACL Meeting, Portland, Oregon, USA, pages 1077–1086, 2011.

[5] D. Yarowsky and G. Ngai, Inducing multilingual PoS taggers and np bracketers via robust projection across aligned corpora, Proceedings of NAACL-

2001 Conference, Pittsburgh, PA, USA, pages 200-207, 2-7 June 2001.

[6] L. Zhu, D.F. Wong, and L.S. Chao, Unsupervised chunking based on graph propagation from bilingual corpus, The Scientific World Journal, Article ID 401943, 2014.

[7] S. Goldwater, T.L. Griffiths, and M. Johnson, Contextual dependencies in unsupervised word segmentation, Proceedings of 21st International Conference on Computational Linguistics and 44th ACL Meeting, Sydney, Australia, pages 673–680, 2006.

[8] D. Mochihashi, T. Yamada, and N. Ueda, Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling, Proceedings of 47th ACL Meeting, Suntec, Singapore, pages 100–108, August 2009.

[9] T. Nguyen, S. Vogel, and N.A. Smith, Nonparametric word segmentation for machine translation, Proceedings of COLING-2010, Beijing, China, pages 815– 823, August 2010.

[10] G. Tambouratzis, M. Vassiliou, & S. Sofianopoulos, Machine Translation with Minimal Reliance on Parallel Resources, Springer Briefs in Statistics series, Springer-Verlag, Berlin, 2017.

[11] E.F. Tjong, K. Sang, and H. Dejean, Introduction to the CONLL-2001 shared task: Clause identification, Proceedings of CoNLL-2001, Toulouse, France, pages 53–57, 6-7 July 2001.

[12] J. Lafferty, A. McCallum and F.C.N. Pereira, Conditional random fields: Probabilistic models for segmenting and labelling sequence data. ICML Conference Proceedings, Williamstown, USA, pages 282–289, 28 June-1 July 2001.

[13] F. Sha and F.C.N. Pereira, Shallow parsing with conditional random fields, Proceedings of HLT-NAACL Conference, Alberta, Canada, pages 213–220, 27 May – 1 Jun 2003.

[14] Y. Tsuruoka, J. Tsujii, and S. Ananiadou, Fast full parsing by linear-chain conditional random fields, 12th EACL Conference Proceedings, Athens, Greece, pages 790–798, 30 March-3 April 2009.

[15] R.O. Duda, P. E. Hart, and D.G. Stork, Pattern Classification (2nd edition), Wiley Interscience, New York, U.S.A., 2001.

[16] G. Tambouratzis, Conditional Random Fields versus template-matching in MT phrasing tasks involving sparse training data, Pattern Recognition Letters, 53:44-52, 2015.

[17] A.J. Booker, J.E. Dennis Jr., P.D. Frank, D.B. Serafini, V. Torczon, and M.W. Trosset, A Rigorous Framework for Optimization of Expensive Functions by Surrogates, Structural Optimisation, 17:1-13, 1999.

[18] Y. Jin, Surrogate-assisted evolutionary computation: Recent advances and future challenges, Swarm and Evolutionary Computation, 1(2):61-70, 2011.

[19] K. Stoeber, P. Wagner, J. Helbig, S. Koester, D. Stall, M. Thomae, J. Blauert, W. Hess, and H. Mangold, Speech synthesis by multi-level selection and concatenation of units from large speech corpora. Verbmobil: Foundations of Speech-to-Speech Translation, Symbolic Computation, Springer, Berlin, pages 519–537, 2000.

[20] J. Kennedy and R. Eberhart, Particle swarm optimization, IEEE International Conference on Neural Networks, Perth, WA, USA, pages 1942–1948, 27 Nov-1 Dec 1995.

[21] G. Tambouratzis, Applying PSO to Natural Language Processing Tasks: Optimizing the Identification of Syntactic Phrases, Proceedings of CEC-2016 Conference, Vancouver, Canada, pp. 1831-1838, July 2016.

[22] R. Mendes, J. Kennedy, and J. Neves, The fully informed particle swarm: Simpler maybe better, IEEE Transactions on Evolutionary Computation, 8(3):204–210, 2004.

[23] Z.-H. Zhan, J. Zhang, Y. Li, and H.S.-H. Chung, Adaptive particle swarm optimization, IEEE Transactions on Systems, Man Cybernetics Part B: Cybernetics, 39(6):1362–1381, 2009.

[24] G. Tambouratzis (2017) The effectiveness of surrogate functions in improving the accuracy of PSO-type algorithms in an NLP task, Proceedings of 10th SSCI Conference, Honolulu, USA, pp. 3214-3221, IEEE Press, 27 November - 1 December 2017.

[25] K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. 40th ACL Meeting Proceedings, Philadelphia, USA, pages 311– 318.

[26] NIST. Automatic evaluation of machine translation quality using n-gram cooccurrences statistics. Report available at http://www.itl.nist.gov/iad/mig/tests/mt/doc/ngramstudy.pdf, 2002.

[27] K.R. Harrison., A.P. Engelbrecht, B.M. Ombuki-Berman, An adaptive particle swarm optimization algorithm based on optimal parameter regions, Proceedings of 10th SSCI Conference, Honolulu, USA, pp. 1606-1613, IEEE Press, 27 November - 1 December 2017.

George Tambouratzis received the Diploma in electrical engineering from the National Technical University of Athens Greece, in 1989, and the M.Sc. degree in digital systems and the Ph.D. degree in neural networks and pattern recognition, both from the Department of Electrical Engineering, Brunel University, U.K., in 1990 and 1993, respectively. Since 1996, he has been associated with the Institute for Language and Speech Processing, Athens, Greece, where he currently is a Senior Researcher. His research interests include neural networks, pattern recognition, evolutionary computation and computational linguistics with emphasis on machine translation. He is a member of the Technical Chamber of Greece and a senior member of the IEEE.
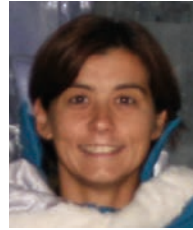
Marina Vassiliou has studied Greek Philology and Linguistics at the University of Athens. She holds a Master's degree in Generative Grammar (University of Athens). Her Master Thesis was titled "The Ancient Greek Infinitive under Control". She has taught various Linguistics courses at Independent Science & Technology Studies/University of Hertfordshire (IST/UH), while for a short period she has worked at the Natural Language Processing Lab of NTUA. In 2003 she translated Noam Chomsky's 2000 book "New horizons in the study of language and mind" (Cambridge University Press). She has published in international journals and conferences. In 2000 she joined the Institute for Language and Speech Processing / R.C. "Athena". Since then she has contributed to various projects concerning statistical approaches to language, stylometry, pattern recognition-based morphological analysis, machine translation, ontologies, controlled languages, specifications for syntactic analysis, coreference resolution.