Journal of World Languages, 2014
Vol. 1, No. 2, 157–170, http://dx.doi.org/10.1080/21698252.2014.937563

Routledge
Taylor & Francis Group

Automatic labelling of transitivity functional roles

Hengbin Yan*

The Halliday Centre for Intelligent Applications of Language Studies, City University of Hong Kong, Hong Kong SAR, China

(*Received 19 May 2014; Accepted 12 Jun 2014*)

Systemic functional linguistics (SFL) is a functionally oriented linguistic framework that has gained increasing influence in recent years, with important applications in the description and analysis of text/discourse. Despite its popularity, relatively little has been done to automate the parsing of functional structures using this framework. Previous attempts have largely depended on non-statistical, rule-based methods, which have limited their application in more complex scenarios. In this article, we present a data-driven method for the classification and labelling of SFL-based functional roles, trained on a recently developed corpus resource. We describe our efforts to engineer lexical, semantic and contextual features in constructing a system for labelling the process types and participant roles in the transitivity system based on the SFL framework. Initial evaluation shows accuracies of 80.5% and 91.8% for the classification of process types and participant roles, respectively. The system is expected to be an important step in achieving fully automated analysis of functional roles in SFL. In addition to applications requiring analysis of English functional structure, we discuss some of the difficulties and future directions in extending the current system to handle less other languages such as Chinese.

Keywords: systemic functional linguistics; functional role labelling; machine learning

1. Introduction

Recent years have seen data-driven approaches to natural language processing successfully applied to a wide range of problems including syntactic (Collins 2003; Klein and Manning 2003), semantic (Gildea and Jurafsky 2002; Pradhan et al. 2004) and discourse (Soricut and Marcu 2003; Hernault, Prendinger, and Ishizuka 2010) analysis. Computational processing of linguistic data for functional analysis using functionally oriented frameworks such as systemic functional linguistics (SFL) (Halliday and Matthiessen 2004), on the other hand, remains a relatively under-explored research area. The functional approach to the description and analysis of the various aspects of language (including structure, meaning and use) has gained increasing influence and adoption as an alternative to formalist theories (Huang 2002). Linguistic analysis using functional theories is still largely manual, and the lack of relevant resources has limited progress in automating the often time- and effort-consuming process.

In this article, I present work being carried out to automate parts of the functional structure using a recently constructed corpus annotated within the framework of SFL (Yan and Webster 2013). Specifically, I focus on the assignment of functional labels in the transitivity system, a core system within SFL which is known to be particularly difficult to process using purely rule-based approaches (Honnibal and Curran 2007). Trained on

*Email: yanhengbin@gmail.com

lexical, syntactic and contextual features obtained from the corpus, automatic procedures are employed to automatically identify and classify the process types as well as accompanying participant roles in unrestricted texts. The proposed system will significantly enhance the potential for applying the SFL framework to the task of automating large-scale text analysis. Finally, I discuss some of the difficulties and future directions, proposing methods such as active learning and annotation projection and extending the current system to handle other languages such as Chinese.

2. Related works

Recent work (Costetchi 2013a, 2013b) has been carried out to automatically generate simplified SFL transitivity parses for unrestricted sentences. Semantic role labelling is performed using a dependency graph-based method and a pattern matching method, resulting in reported accuracy of 72.65% on simple sentences. Although this approach shows respectable results, the current implementation is still largely dependent on hand-crafted patterns that limit its application to more complex, unrestricted texts.

To address the bottlenecks of current state-of-the-art functionally based parsers, a small-scale corpus (Yan and Webster 2013) has recently been annotated with SFL functional roles. The annotation of the corpus was done in four successive layers:

Clausal: clausal boundaries, including boundaries of embedded clauses. The clause boundaries are aligned with the RST Treebank where clausal boundaries are also annotated.

Process: processes are the core of a clause, typically realized by a verbal group headed by the root verb of the clause. As described in Halliday (1994), there are 6 common types of processes (material, behavioural, mental, verbal, relational and existential), subdivided into 10 more refined types (with *material* subdivided into *doing, happening, mental* into *perception, cognition, affection* and *relational* into *attributive, identifying*). Each of the process types is associated with a set of nuclear and non-nuclear participants.

Participant: participants are the central nominal groups of the clause typically realized by the grammatical Subject or Object of the clause. A summary of the processes with its related participants is shown in Table 1.

Circumstance: peripheral units related to time, place, manner, etc. typically realized by adverbial groups. There are in total nine broad types of circumstances: *Extent, Location, Manner, Cause, Contingency, Accompaniment, Role, Matter* and *Angle*, each with its own subtypes. The *Extent* circumstance, for example, is subdivided into three subtypes: *duration, frequency and distance*.

Table 1. A summary of the process types and participants in the transitivity system.

Process type	Nuclear participants	Example
Material	Actor, Goal	$\text{She}_{\text{Actor}}$ made the $\text{coffee}_{\text{Goal}}$
Mental	Senser, Phenomenon	$\text{She}_{\text{Senser}}$ saw the $\text{car}_{\text{Phenomenon}}$
Attributive	Carrier, Attribute	$\text{Maggie}_{\text{Carrier}}$ was $\text{strong}_{\text{Attribute}}$
Identifying	Identified, Identifier	$\text{Maggie}_{\text{Identified}}$ was our $\text{leader}_{\text{Identifier}}$
Behavioural	Behaver, (Target)	$\text{She}_{\text{Behaver}}$ laughed
Verbal	Sayer, (Target)	She replied
Existential	Existent	There was a beautiful $\text{princess}_{\text{Existent}}$

In total, 81 documents from the Wall Street Journal section of the Penn Treebank have been annotated, with a total of 43,351 words, divided into 1621 sentences and 4620 clauses. The corpus, though still relatively small-scale, serves as the basis for further statistical modelling and supervised training in our proposed system. In the following sections, I describe the procedure for constructing such a data-driven system.

3. Classification

The focus of this article is on the system constructed for labelling functional roles in the transitivity system (Table 1). In an SFL-based analysis, three strands of meaning (called metafunctions) operate concurrently: the *ideational (experiential* and *logical), interpersonal (social interaction)* and *textual (communicative organization)* metafunctions. The *transitivity* system is a major system in the *ideational* system. Specifically, it involves a configuration of processes and participants involved (such as *Actor, Goal*) and the accompanying circumstances (such as *time, place, manner*). The task of labelling transitivity roles is divided into two steps: (1) clause boundary identification and (2) transitivity role classification.

Clause boundary identification includes the task of dividing texts and sentences into clauses, or Elementary Discourse Units (EDU). Identification of clause boundaries is usually the first step in a functional analysis. Although considered a kind of lower-level segmentation, clause boundary identification is crucial to the quality of further discourse parsing. Recent advances in discourse parsing (HILDA) have yielded good results in clause segmentation, achieving an F-score of 93.8%, or 96% of the human performance level (Hernault, Prendinger, and Ishizuka 2010). Our role labelling system uses the state-of-the-art discourse segmenter provided by the HILDA parser for clause boundary identification.

On the other hand, the second step of classifying transitivity roles is less well developed. We may approach the problem first as one of disambiguating the different senses of the process as realized by a verb, and second as a Semantic Role Labelling problem whereby the participant roles associated with the process are identified. Transitivity analysis begins with identifying the process and process type in the clause. For example, in the clause *John hit the ball*, when the type of the process *hit* is identified as *material*, the related participant roles of *John* and *the ball* can be inferred from their grammatical functions (*John* being the grammatical Subject and *the ball* being the grammatical Object) in the clause. The step may be divided into two subtasks: (1) classification of the process type and (2) identification of the participant roles based on the identified process type.

3.1 Process types

Grammatically, the process is a verbal group with the verb as its core. We can simplify the task by reducing it to disambiguating the sense of the core verb. For an arbitrary input sentence, we first segment it into clauses using the technique used by a clause boundary identifier as described in Hernault, Prendinger, and Ishizuka (2010). Features from each clause and its contexts are then extracted for classification.

3.1.1 Features

We employ a combination of lexical, syntactic and semantic features as shown in Table 2.

Table 2. The features used in classification of process types and the scope that these features are extracted.

Features	Scope
Root verb in the clause	Clause
Subject (if any) of the root	Sentence
Object (if any) of the root	Sentence
Lemmatized form of the root verb	Lexicon
WordNet lexical name of the process and participants disambiguated by sense disambiguation systems	External

3.1.2 *Syntactic features*

An input text is first segmented into individual sentences using a sentence segmenter (Bird 2009). For every sentence, we obtain the Stanford Typed Dependencies (STD) from the Stanford Parser (Klein and Manning 2003; De Marneffe, Maccartney, and Manning 2006) and extract the root verb of each clause and the corresponding syntactic Subjects and Objects (typically also serving as functional participant roles in the clause).

The STD is a binary representation of the relations among words in a parsed sentence. It retains basically the same syntactic information as a phrase structure parse, but provides a straightforward format that makes it easier to extract dependency relationships. Each dependency relation in the STD is represented by a triplet of (1) the relation, (2) the governor (or regent/head) and (3) the dependent. The current representation includes 53 relations organized hierarchically. For example, the STD for the sentence *The Court's decision will have billion-dollar consequences for manufacturers* is as follows:

```
det(Court-2, The-1)
poss(decision-4, Court-2)
nsubj(have-6, decision-4)
aux(have-6, will-5)
dobj(have-6, consequences-8)
amod(consequences-8, billion-dollar-7)
prep_for(consequences-8, manufacturers-10)
```

Each line represents a binary dependency relation between two words. The first symbol (e.g. det) is the name of the relation, followed in the bracket by the governor (e.g. Court-2, 2 being the index) and the dependent (e.g. The-1).

The dependency structure of a sentence allows us to extract syntactic features available in the sentential parse. For example, in the example sentence above, it is straightforward to identify the verb *have* as the root of the sentence. The root *have* has three direct dependents: *decision, will* and *consequences*. For the dependency relations of the direct dependents (*nsubj, aux, dobj*) with the root verb, we deduce their grammatical roles in the sentence: Subject, Auxiliary and Object. Each of the direct dependents of the root can have its own dependent. For example, the noun *decision* is a governor of the noun *Court*, which in turn governs the determiner *The*. Using the STD makes it straightforward to extract features both at the sentence and lexical levels.

3.1.3 Feature extraction

In the following, we consider the feature building process with an example sentence:

If that controversy continues, // other foreign producers are likely // to grab most of the sales in Eastern Europe.

The sentence is a complex sentence that can be divided into three clauses as follows (each clause separated by //). The identification of clause boundaries is done with the clause segmenter as described in Hernault, Prendinger, and Ishizuka (2010).

To build the features, we first obtain the parsed STD from the Stanford Parser:

```
mark(continues-4, If-1)
det(controversy-3, that-2)
nsubj(continues-4, controversy-3)
advcl(are-9, continues-4)
amod(producers-8, other-6)
amod(producers-8, foreign-7)
nsubj(are-9, producers-8)
root(ROOT-0, are-9)
acomp(are-9, likely-10)
aux(grab-12, to-11)
xcomp(likely-10, grab-12)
dobj(grab-12, most-13)
det(sales-16, the-15)
prep_of(most-13, sales-16)
nn(Europe-19, Eastern-18)
prep_in(sales-16, Europe-19)
```

In this example, we first identify the root of the sentence (*are*). The process in each of the clauses is the verbal group dependent on and having the shortest distance from the root of the sentence. For example, the verb *continues* in the first clause is directly dependent on the root verb *are* and thus has a dependency distance of 1, the shortest of all the words in the clause. The root verb is then lemmatized to its base form using the WordNet Lemmatizer (Bird 2009). For example, the plural copula *are* is lemmatized to *be* (Tables 3 and 4).

The nominal groups realizing the grammatical roles of Subject and Object in each clause are then derived from the STD. The STD provides basic semantic role labelling

Table 3. Process information related to the example clause.

Clause	Process	Lemmatized process (event)	Distance from root
If that controversy continues,	continues	continue	1
other foreign producers are likely	are	be	0
to grab most of the sales in Eastern Europe.	grab	grab	1

Table 4. Verbal group structure for “has not been eating”.

has	not	been	eating
Finite	Polarity	Auxiliary	Event

deduced from the parsed syntactic structure of a sentence, indicated by their relations with the verb, allowing us to identify the Subject (or Agent if the clause is in the passive voice), the Direct Object and the Indirect Object in a clause. In most cases, the Subject and Object of a process are within the same clause. It is worth noting, however, that sometimes the Subject and Object are absent from the clause and can only be deduced at the sentential level. In the third clause of the above example, the Object of the third clause is identified as *most of the sales in Eastern Europe*, but the Subject of the verb *grab* is missing from the clause itself. We deduce from the dependency chain that the first participant for the process *grab* is also *producers*. The *nil* placeholder is used to indicate cases where the Subject/Object is not found either in the clause or in the sentence (Table 5).

Identifying the grammatical roles associated with the process provides useful information about the participants. The presence or absence of the Object informs as to whether the process/verb is being used transitively or intransitively. The other lexical semantic properties of the grammatical roles can also be useful in distinguishing the various types of processes. Apart from syntactic features, we also consider external resources that may be useful in helping to distinguish process types, such as WordNet (Miller 1995). In WordNet, words are grouped into cognitive synonyms called synsets. The synsets in WordNet are organized into 45 lexicographer files that divide words (mainly nouns and verbs) into semantic categories (Table 6).

There is an interesting correspondence between the WordNet lexicographer categories and the process types in SFL. For example, the categories *verb.cognition, verb.emotion* and *verb.perception* correspond to the mental process types of *cognition, affection* and

Table 5. Grammatical role features for the example clauses.

Clause	Process	Subject	Direct Object	Indirect Object
1	*continues*	*controversy*	nil	nil
2	*are*	*producers*	*likely*	nil
3	*grab*	*producers*	*sales*	nil

Table 6. WordNet's lexicographer categories for verbs.

Lexicographer file name	Meaning
verb.body	verbs of grooming, dressing and bodily care
verb.change	verbs of size, temperature change, intensifying, etc.
verb.cognition	verbs of thinking, judging, analysing, doubting
verb.communication	verbs of telling, asking, ordering, singing
verb.competition	verbs of fighting, athletic activities
verb.consumption	verbs of eating and drinking
verb.contact	verbs of touching, hitting, tying, digging
verb.creation	verbs of sewing, baking, painting, performing
verb.emotion	verbs of feeling
verb.motion	verbs of walking, flying, swimming
verb.perception	verbs of seeing, hearing, feeling
verb.possession	verbs of buying, selling, owning
verb.social	verbs of political and social activities and events
verb.stative	verbs of being, having, spatial relations
verb.weather	verbs of raining, snowing, thawing, thundering

perception. Categories such as *verb.possession* and *verb.stative* correspond to relational processes of *attribution* and *identification*. Other categories such as *verb.body* or *verb.competition* are less directly correlated, but may roughly correspond to certain process types. On the basis of such correspondences, we discover valuable clues for the classification of process types.

Since a given word may belong to more than one category, determining the categories in context requires us to disambiguate the senses of the word. To do this, we take advantage of existing state-of-the-art methods for Word Sense Disambiguation (WSD). Agirre and Soroa (2009) and Agirre, De Lacalle, and Soroa (2013) present a graph-based method (*UKB*) that uses a Lexical Knowledge Base to perform unsupervised WSD which has achieved performance comparable to state-of-the-art supervised techniques. We use UKB to perform a WSD task whereby we determine the sense of the content words in a sentence as well as the lexicographer categories as indicated in the WordNet database. For each process and grammatical role in a clause, we determine its lexicography category in WordNet. To do this, we feed the process and its context (content words in the same clause as the process) in the sentence into UKB to be disambiguated. Once the word sense as represented by a WordNet synset is disambiguated, we look up the lexicography category in WordNet. The result for the example sentence is shown as follows (Table 7):

As indicated by the criteria for distinguishing process types as described in Halliday (1994), the nature of participants (whether they are an unconscious thing, or a conscious thing or a fact) may help distinguish the type of the process. These lexicographer categories provide such useful information as whether a participant is an animate entity, a person, a social group (a group of persons) or a fact. In the example above, UKB correctly identifies the category for *producer* as *noun.person*.

3.2 Participants

The configurations of process types and their participants are relatively fixed. The choice of process types in the system comes with a set of participant roles that are typically involved in the process. For example, a verbal clause is characterized by a configuration of *Process (obligatory) + Sayer (obligatory) + Verbiage (optional) + Receiver (optional)*. In the previous section, we used a number of features to predict the type of the process. Given the process type, and the potential configurations of participant roles involved in the process, we should be able to predict the participant roles in the clause.

Participants in SFL are typically realized structurally by nominal groups. By identifying the nominal groups in a clause, we can also identify candidates for being assigned participant roles. However, not all nominal groups have participant roles. Functionally,

Table 7. Lexical information for functional constituents in the example clauses.

Lemma	Number of WordNet senses	UKB-disambiguated WordNet synset	WN lexicographer category for Lemma
continue	10	continue.v.01	verb.stative
be	13	be.v.01	verb.stative
grab	6	catch.v.04	verb.contact
controversy	1	controversy.n.01	noun.communication
producer	3	manufacturer.n.02	noun.person
sales	5	sale.n.03	noun.act

only those that are centrally involved with the process are considered participants. For example, in the clause *I had a dream last night, I* and *a dream* are centrally involved with the process *had*, and are therefore participants, while *last night* serves as the more peripheral *circumstance*. Syntactically speaking, a nominal group has to both be directly dependent on the process (realized as the root verb of the clause) and also form certain types of dependency relations with the process to be considered a participant.

There are a few problems to consider for the configuration of participant roles. The first is the grammatical roles the participants serve in a clause. In traditional grammar, these roles are referred to as the Subject and Object. In SFL, they are functional constituents called Subject and Complement in the Mood System. In SFL, Subject and Complements are typically identified using certain grammatical tests such as the tag set and plurality test (Eggins 2004), which allude to humans' inherent linguistic knowledge and are difficult to perform computationally. Instead, we take advantage of syntactic relations obtained from syntactic parsers. Table 8 shows some STD relations that can be used to infer the grammatical relations among the participants such as whether the nominal is a Subject (e.g. nsubj) or Object (e.g. dobj, iobj). By using such syntactic information, we can identify the grammatical roles (Subject, Direct Object and Indirect Object) of a clause. The second problem is identifying the semantic roles of the nominal group in the Subject or Object positions. In an active clause like *the car hit the wall*, the Subject of the clause (*the car*) is conflated with the *Agent/Actor* of the clause. In a passive clause, however, the Subject is conflated with the *Patient/Goal* of the clause. In typed dependencies, a distinction is also made in passive clauses between the passive Subject (i.e. *nsubjpass, csubjpass*) and the real *Agent*, typically in a prepositional phrase headed by the preposition *by*. In certain cases, clauses with verbs like *give* may have more than one Object, as in *He gave a pen* (Direct Object) *to her* (Indirect Object). Again information about both types of Objects is available from the syntactic parses rendered as typed dependencies. Such syntactic information and the process type of the clause allow for the extraction of features to construct another classifier for the participant roles in the clause. We determine the participant roles of each of the present Subject/Object using the features in Table 9.

For example, the following features are extracted for the example sentence *Campeau operates department store chains and is strained for cash.* (Table 10):

The process type is obtainable from the earlier classifier that we have built for process type classification. It is worth noting that although the number of participant roles present

Table 8. Common types of dependency relations of arguments that can serve as the Subjects or Complements of a clause.

Dependency relation with the root	Meaning	Grammatical Role/example participant role
nsubj	Nominal Subject	Subject/Actor
csubj	Clausal Subject	Subject/Actor
nsubpass	Passive Nominal Subject	Subject/Actor
csubjpass	Passive Clausal Subject	Subject/Goal
agent	Agent	Object/Goal
dobj	Direct Object	Object/Goal
iobj	Indirect Object	Object/Recipient
ccomp	Clausal Complement	Object/Goal
expl	Expletive	Subject/nil
prep_that	Prepositional Modifier	Object/Goal

Table 9. Features used for the classification of participant roles.

Feature ID	Feature description
1	Process type of the clause the participant is in
2	The dependency relation of the participant with the root verb
3	Whether the participant is in the Subject position of the clause
4	The position of the participant relative to other participants (can be first, second or third)

Table 10. Features for classifying participants in the example clause.

Participant	Process type	Dependency	Is Subject?	Position
Campeau (c. 1)	doing	nsubj	Yes	1st
chains (c. 1)	doing	dobj	No	2nd
Campeau (c. 2)	doing	nsubjpass	Yes	1st

in the corpus is 18 in total, in fact only a few choices are available given a particular process type. For the material process type, for example, the available participant roles are the following five: *Initiator, Recipient, Client, Scope* and *Attribute*.

4. Evaluation and performance

The data from our corpus are divided into a tuning set and an evaluation set. We use 20 of the 81 documents for feature engineering and model development, and the remaining documents for evaluation, performing a 10-fold cross-validation on a random forests classifier (Breiman 2001). Random forests are an ensemble classifier based on decision trees. The idea is that growing an ensemble of decision trees and letting them vote in a classification task can lead to significant improvement in accuracy. The theoretical underpinnings for the learning model are detailed in Breiman (2001) and Liaw and Wiener (2002). The advantages of random forests over other learning algorithms are that (1) it is an effective tool in prediction that yields state-of-the-art performance, (2) it does not overfit, (3) it is fast, (4) it gives estimates of the relative importance of variables (useful in feature selection) and (5) it is effective in estimating missing data (such as the participant roles which are often missing in our feature sets). We also compare the performance of the random forests classifier with another state-of-the-art classifier based on Support Vector Machine (SVM) (Chang and Lin 2011). We use a simple baseline to compare with the performance of the classifier. The baseline performance is obtained by always choosing the most frequent class in the test data. In our corpus, the most frequent process type is *doing*, and thus, the baseline classifier would just pick this as the predicted class without considering any other features (Table 11).

Table 11. Performance statistics for the classification of process types. The statistics in bold indicate the best performance among the sets of features.

Features	Model	Precision	Recall	F-Measure
Naïve Baseline	RF	0.199	0.447	0.275
Verb Only	RF	0.731	0.760	0.745
Verb + Syntactic	RF	0.769	0.780	0.774
Verb + Syntactic + UKB	RF	**0.803**	**0.808**	**0.805**
Verb + Syntactic + UKB	SVM	0.773	0.751	0.762

The classification results seem to suggest that the classification of process types is still heavily lexically grounded. Most of information for the classification can be deduced from looking at the word itself and guessing its possible function without taking other contextual features such as syntactic dependencies and surrounding content words into account. On the other hand, the information as encoded in the syntactic and contextual features conflates with the lexical information to a large degree. This is reflected from our observation that even without using the words themselves as features we still get more or less the same level of performance. As a result of the conflation, although the syntactic and WSD features have proven useful in disambiguating the process types, the performance increase is not as great as expected. To some degree, this is not surprising since the disambiguation of functions, with its similarity to WSD, is an AI hard problem, and only a few WSD systems have managed to achieve marginal gains over the most frequent sense on "all-words exercises" (as opposed to limiting the number of words to only a handful) despite years of research into the problem.

Looking at class-specific performance (Table 12), we see that the *behavioural* process has the lowest accuracy. This conforms with feedback from corpus annotators during the annotation phase when they complained about the difficulties in differentiating the *behavioural* from other types of process such as *material:doing* and *mental*. The second worst performing process type is *affection*. This might be explained by its low number of instances in the testing corpus, since the performance of a class can be affected adversely by insufficient training samples. The *affection* process type is followed by *identifying* and *happening*, both of which are subcategories of a major process type (*relational* and *doing*) and may be more easily confused with their close "relatives" (*attributive* and *doing*). The class with the highest accuracy is *verbal*, presumably due to there being only one class in the major process type and there are relatively few ambiguous word types that realize verbal functions (e.g. most commonly *say, state, announce*). Another evaluation is performed on the classification of participant roles. Similar to process type evaluation, we also set up a naïve baseline that always picks the most frequent participant role in the corpus. We also compare the performance of the classifier using a few ablations of the features (the numbers refer to the feature ID in Table 9).

As seen from Table 13, the random forest classifier has yielded relatively high performance in classifying participant roles. Starting with only the process type as features, the performance is poor. However, with the addition of grammatical role information – such as whether the participant is in the Subject position – the most substantial gain in performance is achieved. This confirms our intuition that knowing the process type of a clause may put the participants in the right functional configuration,

Table 12. Class-specific performance for each process type using random forests.

Class	Precision	Recall	F-Measure
verbal	0.905	0.882	0.893
doing	0.819	0.907	0.860
attributive	0.857	0.857	0.857
cognition	0.813	0.777	0.794
behavioural	0.324	0.190	0.240
affection	0.882	0.556	0.682
perception	0.870	0.690	0.769
existential	0.759	0.688	0.721
identifying	0.731	0.667	0.698
happening	0.767	0.646	0.701

Table 13. Performance statistics for classification of participant roles. The numbers (1), (2), (3), (4) refer to the feature ID in Table 9. The statistics in bold indicate the best performance among the sets of features.

Features	Model	Precision	Recall	F-Measure
Naïve baseline	RF	0.095	0.308	0.145
(1) only	RF	0.562	0.646	0.601
(1) + (3)	RF	0.853	0.877	0.865
(1) + (2) + (3) + (4)	RF	**0.907**	**0.929**	**0.918**
(1) + (2) + (3) + (4)	SVM	0.897	0.924	0.910

Table 14. Detailed performance by class for classification of participant roles.

Class	Precision	Recall	F-Measure
Phenomenon	0.899	0.879	0.889
Senser	0.924	0.937	0.931
Sayer	0.977	0.991	0.984
Verbiage	0.810	0.823	0.816
Value	0.975	0.994	0.984
Token	0.993	0.972	0.982
Goal	0.888	0.956	0.920
Actor	0.944	0.934	0.939
Carrier	0.994	0.969	0.981
Attribute	0.945	0.992	0.968
Recipient	0.333	0.014	0.028
Receiver	0.000	0.000	0.000
Existent	1.000	1.000	1.000
Client	0.000	0.000	0.000
Scope	0.766	0.901	0.828
Beneficiary	0.000	0.000	0.000
Behaver	0.906	0.935	0.921
Behaviour	0.000	0.000	0.000

and information on grammatical relations within the configurations can be crucial in disambiguating the participant roles (Table 14).

The detailed performance across the classes seems to be quite divided, ranging from perfect prediction to zero accuracy. The class that achieved 100% accuracy is the relatively simple case of *Existent*, which is the only possible participant given the existential process. The overall performance for the nuclear participants is high, while the worst performing classes are all non-nuclear participants (*Receiver, Client, Beneficiary, Behaviour, Recipient*). This may have been due to the low percentage of the non-nuclear participants as well as the inability of the feature for the relative position of the grammatical roles to distinguish between nuclear and non-nuclear participants. Overall, while there is still room for further improving the accuracy of the transitivity classifier, initial experiments have shown promising results and support the notion that automating transitivity analysis is both practical and feasible using data-driven machine-learning techniques.

5. Extending to resource-poor languages

So far our discussion has been limited to the labelling of transitivity structure in the English language. Compared with English with its lexical resources (e.g. WordNet),

annotated corpora (e.g. Penn Treebank), syntactic parsers (e.g. the Stanford Parser) and semantic analysis tools (e.g. UKB), available linguistic resources in most other languages are scarce in comparison. For example, Chinese, while being one of the most widely used languages, has a relative paucity of lexically and semantically annotated resources, and it is only in recent years that efforts have been made to build Chinese counterparts of existing English corpora and tools (Li et al. 2003; Xue et al. 2005). Given this resource scarcity, extending our system to deal with languages other than English is a challenging task. Despite this, we propose to overcome the problems by adopting two general approaches.

The first is *active learning* (Zhu, Lafferty, and Ghahramani 2003) in the annotation process. Our current corpus has been annotated mostly manually, with the help of a web-based collaborative platform. The uneven distribution of words and their senses often means that most of the efforts go into annotation of frequently appearing words, which is often unnecessarily repetitive and adds little value to subsequent disambiguation of semantic senses. Active learning is where a learning algorithm tells us which set of unlabelled data to label. This is often more desirable than selecting our own set of data to label randomly. In addition to expanding the current functional corpus, active learning is also particularly useful when creating new functional resources in a new language such as Chinese, substantially reducing the time and cost involved in the annotation.

The second is the use of *annotation projection* (Pad and Lapata 2009). The key idea behind annotation projection is that, given a pair of parallel corpora that are translations of each other, one in English (E) and one in a less resource-rich language such as Chinese (C), we can first annotate E for its functional structure and then project this functional structure onto C by relying on word alignment information in the translation pairs. After a sizable set of the sentences in C has been annotated with functional structure through annotation projection, a classifier can then be trained on the set independently of the parallel corpora.

With the use of semisupervised learning and annotation projection, we envision that the extension of our current system to cover resource-poor languages will be made significantly more efficient.

6. Conclusion

In this article, I have discussed the problem of automating the analysis of SFL transitivity function structures. A small-scale corpus was annotated for its transitivity structure to which has been applied state-of-the-art machine-learning algorithms for the automatic classification of the process types and participant roles of clauses. I modelled the tasks as a role labelling task and a sense disambiguation task, detailing the process in which we engineered key features for the classification. The labelling system was able to achieve performance significantly better than our baselines.

Due to difficulties in automatic text analysis using SFL, current state-of-the-art work is still limited both in scope and in functionality. As with other semantically oriented tasks in NLP such as semantic role labelling, determining the multifaceted functional roles of a functional component is an AI hard problem, more complicated than their syntactic counterparts as this depends more on the configuration of a wide range of contextual factors such as the functional semantic configuration in different usage patterns. Due to such difficulties, a parser for producing the complete SFL structures of free-form texts has yet to be developed. What is proposed here is a new perspective inspired by advances in other fields of NLP which have achieved success. Work is ongoing in The Halliday Centre

at City University of Hong Kong to build on this system to create a more comprehensive and robust parser for SFL analysis.

Notes on contributor

Hengbin Yan is a Postdoctoral Fellow at The Halliday Centre for the Intelligent Applications of Language Studies, City University of Hong Kong. He received his Ph.D. in Computational and Functional Linguistics from City University of Hong Kong, and his M.A. and B.A. from Guangdong University of Foreign Studies. He has worked both within and outside academia, holding positions such as Senior Research Associate and Research Engineer. His main research interests include Systemic Functional Linguistics, Computational Linguistics, and Corpus Linguistics. He is particularly interested in applying computational and corpus-based techniques to find innovative solutions to difficult problems in linguistic theories.

References

Agirre, Eneko, and Aitor Soroa. 2009. "Personalizing Pagerank for Word Sense Disambiguation." In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, 33–41. Athens, Greece: Association for Computational Linguistics.

Agirre, Eneko, Oier Lopez de Lacalle, and Aitor Soroa. 2014. "Random Walks for Knowledge-Based Word Sense Disambiguation." *Computational Linguistics* 40 (1): 57–84.

Bird, S. 2009. *Natural Language Processing with Python*. 1st ed. Beijing: O'Reilly.

Breiman, L. 2001. "Random Forests." *Machine Learning* 45 (1): 5–32. doi:10.1023/A:1010933404324.

Chang, C.-C., and C.-J. Lin. 2011. "LIBSVM: A Library for Support Vector Machines." *ACM Transactions on Intelligent Systems and Technology (TIST)* 2 (3): 27.

Collins, M. 2003. "Head-Driven Statistical Models for Natural Language Parsing." *Computational Linguistics* 29 (4): 589–637. doi:10.1162/089120103322753356.

Costetchi, Eugeniu. 2013a. "A Method to Generate Simplified Systemic Functional Parses from Dependency Parses." In *DepLing 2013*, 68–77. Prague, Czech Republic: Charles University in Prague.

Costetchi, Eugeniu. 2013b. "Semantic Role Labelling as SFL Transitivity Analysis." In *ESSLLI Student Session 2013 Preproceedings*, 29–39. Düsseldorf, Germany: Heinrich Heine Universität in Düsseldorf.

De Marneffe, Marie-Catherine, Bill MacCartney, and Christopher D. Manning. 2006. "Generating Typed Dependency Parses from Phrase Structure Parses." In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, 6:449–54. Genoa, Italy: European Language Resources Association.

Eggins, S. 2004. *An Introduction to Systemic Functional Linguistics*. New York: Continuum.

Gildea, D., and D. Jurafsky. 2002. "Automatic Labeling of Semantic Roles." *Computational Linguistics* 28 (3): 245–288. doi:10.1162/089120102760275983.

Halliday, M. A. K. 1994. *An Introduction to Functional Grammar*. 2nd ed. London: Edward Arnold.

Halliday, M. A. K., and C. M. Matthiessen. 2004. *An Introduction to Functional Grammar*. London: Edward Arnold.

Hernault, Hugo, Helmut Prendinger, and Mitsuru Ishizuka. 2010. "HILDA: A Discourse Parser Using Support Vector Machine Classification." *Dialogue & Discourse* 1 (3): 1–33.

Honnibal, M., and J. R. Curran. 2007. "Creating a Systemic Functional Grammar Corpus from the Penn Treebank." In *Proceedings of the Workshop on Deep Linguistic Processing*, 89–96. Prague, Czech Republic: Association for Computational Linguistics.

Huang, G. 2002. "Hallidayan Linguistics in China." *World Englishes* 21 (2): 281–290. doi:10.1111/1467-971X.00248.

Klein, Dan, and Christopher D. Manning. 2003. "Accurate Unlexicalized Parsing." In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, 423–30. Stroudsburg, PA, USA: Association for Computational Linguistics.

Li, M., J. Li, Z. Dong, Z. Wang, and D. Lu. 2003. "Building a Large Chinese Corpus Annotated with Semantic Dependency." In *Proceedings of the Second SIGHAN Workshop on Chinese*

Language Processing-Volume 17, 84–91. Sapporo, Japan: Association for Computational Linguistics.

Liaw, A., and M. Wiener. 2002. "Classification and Regression by Random Forest." *R News* 2 (3): 18–22.

Miller, G. A. 1995. "WordNet: A Lexical Database for English." *Communications of the ACM* 38 (11): 39–41. doi:10.1145/219717.219748.

Pad, S., and M. Lapata. 2009. "Cross-Lingual Annotation Projection for Semantic Roles." *Journal of Artificial Intelligence Research* 36 (1): 307–340.

Pradhan, Sameer S., Wayne Ward, Kadri Hacioglu, James H. Martin, and Daniel Jurafsky. 2004. "Shallow Semantic Parsing Using Support Vector Machines." In *HLT-NAACL*, 233–40. Boston, MA: The Association for Computational Linguistics. http://acl.ldc.upenn.edu/hlt-naacl2004/main/pdf/133_Paper.pdf.

Soricut, Radu, and Daniel Marcu. 2003. "Sentence Level Discourse Parsing Using Syntactic and Lexical Information." In, *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, *1*, 149–56. Edmonton, Canada: Association for Computational Linguistics.

Xue, N., F. Xia, F.-D. Chiou, and M. Palmer. 2005. "The Penn Chinese Treebank: Phrase Structure Annotation of a Large Corpus." *Natural Language Engineering* 11 (2): 207–238. doi:10.1017/S135132490400364X.

Yan, H., and J. J. Webster. 2013. "A Corpus-Based Approach to Linguistic Function." In *Proceedings of the 27th Pacific Asia Conference on Language, Information, and Computation (PACLIC 27)*, 215–221, Taipei, Taiwan: Association for Computational Linguistics and Chinese Language Processing.

Zhu, Xiaojin, John Lafferty, and Zoubin Ghahramani. 2003. "Combining Active Learning and Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions." In *ICML* 2003 *Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 58–65. Washington, DC: AAAI Press.