XTR-004: An algebra of concepts, verbs and overlays

Ladislau Bölöni
Dept. of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816–2450
lboloni@eecs.ucf.edu

(XTR-004)

***Abstract*—**
The Xapagy cognitive architecture relies on objects which are characterized with overlays of concepts and verbs. This report describes the reasoning behind the existence of this framework, by describing the affordances which the framework is intended to provide for the system. These affordances are discovery, change, matching, reference, learning and externalization.
The system of concepts, verbs and overlays implement these affordances by relying on a specific algebra which manipulates these entities. We describe the mathematical forms of the operations defined on these entities. Finally, through a series of examples relying on the artificial domain, we illustrate the practical results of these operators.

I. Introduction

The two main objects of the Xapagy architecture are instances (roughly mapping to entities in the real world) and verb instances (VIs) roughly mapping to events, actions and relations between entities. These entities are characterized by a set of *attributes*. In order to achieve its goals, the cognitive architecture put certain requirements on the attributes of instances and VIs. We translate these requirements into a set of *affordances*, which must be satisfied by the attribute framework. For example, the attributes must allow the approximate matching of the instances againsts each other, or the learning of new types of attributes.

An attribute system consisting of a series of discrete labels can not satisfy these affordances – we need a more complex representation. The approach chosen in Xapagy relies on the attributes of instances being represented by *overlays of concepts*, while the attributes of VIs are represented by *overlays of verbs*. The individual concepts and verbs themselves have a specific structure allowing for overlaps and impacts. Overlays of both can be built, changed and compared using specific mathematical operators. These structures and operators add up to an *algebra of concepts, verbs and overlays*.

A. *Affordances required of the instance attributes*

Xapagy instances represent entities of the real world, over the span of a certain time - in general we assume that an entity is represented by the same instance, as long as it does not change "too much".

The properties of a certain instance are described by its *attributes*. The concept and concept overlay mechanism is designed a structure in the attributes of the instance. The structure is designed in such a way that it provides a series of *affordances* of the attribute system. We will concentrate on the following affordances: Construction, Discovery, Change, Matching, Reference and Learning.

Construction: this affordance allows the construction of an instance with a specific set of attributes. One specific challenge here is the resolution of conflicting attributes.

Discovery: is the affordance through which an instance can acquire new attributes without loosing the previous ones. It can be conceptualized as an external observer discovering new attributes by observing the entity.

Change: allows the entity to change some of its attributes in a way which is not compatible with discovery. One of the simplest examples is the case when the entity looses one or more previously existent attributes. In Xapagy, this means that a new instance is created with a new set of attributes. The new instance will have some of the attributes related to the previous one, and the transfer of some of the attributes from the old instance to the new one must be specified.

Matching: is the affordance of being able to determine the degree of matching between two instances in the sense of sharing certain attributes. The degree of matching needs to follow the intuition that instances are more closely matched if their commonalities are rare: eg. the matching between two humans called Achilles is closer than between two instances which have in common the fact that they are both Greeks.

Reference: allows us to determine whether a set of attributes can be used to refer to a given instance. Alternatively, we can say that this affordance enables selection from a set of instances by a reference attribute. Reference combines matching with measurement of incompatibility. For instance Achilles and Ulysses match quite closely, being both Greek warriors. However, only one of them can be referred with the attribute Achilles.

Learning: this affordance allows new types of attributes to be added to the system and the adjustment between the relationships of existing attributes, preferably with minimal disruption towards the other affordances.

This list of affordances misses a number of structures which are considered important in other knowledge representation systems, such as hierarchies, types, generalization / specialization and the class / instance dichotomy. These structures are part of the human model of dealing with knowledge, but they are not considered foundational in Xapagy. If their semantics is needed, they must be constructed on top of the existing model.

B. *Affordances required of the VI attributes*

The attributes of VIs also require a number of affordances, which are similar to the ones we consider for instances.

Construction: this affordance allows a particular VI to be constructed with a specific set of attributes.

Matching: this affordance allows the VIs to be compared for similarity of their attributes. The main application of these affordances is in shadowing. Intuitively, the affordance should reward common attributes which are rare. Two VIs whose only common attribute is that they both represent an action are very weakly matched, however, if both of them represent a "cutting" operation, they will have a higher level of matching.

Learning: this affordance allows new types of attributes to be added to the VIs. This allows us, for instance, to introduce new types of events and actions from direct sensing which had not been witnessed before. Another application is the development of internal representations of unknown words in language learning. Finally, another important application is the representation of summarization verbs which summarize frequently encountered story snippets.

The verb framework does not require the reference and the change affordances, as VIs are not changed over the course of the operation of the system, and the can not be referred to directly.

C. Organization of this report

This report describes the algebra which underlies the concept, verb and overlay system. The goal is to illustrate how this framework implements the affordances mentioned above. This report does not deal with the learning affordance, which will be discussed in a future report.

Section **??** describes the basic properties of concepts and verbs, and the structure and building of overlays. These operations cover the construction and change affordances. Section II describes the binary relations on overlays which form the basis of the implementation of matching and reference affordances. Finally, Section IV describe a series of experiments illustrating the functionality of these operators over examples of concept and verb overlays based on the artificial domain.

II. Properties of concepts and verbs. Building overlays

A. Concepts and concept overlays

A *concept* in Xapagy is the representation of an *undivisible attribute*. A weighted superposition of concepts is called a *concept overlay* (CO). When talking about concepts in general, we will denote concepts with $c_1, c_2 \dots$ and COs with $C_1, C_2 \dots$. For specific concepts we will use descriptive names in brackets such as `[man]`. For specific overlays, we list the participating concepts inside brackets, if necessary, specifying the explicit energy level of each concept in the overlay.

The *specificity* of a concept is characterized by its *area:* $area(c) \in \mathbb{R}^+$. The more specific a concept is, the smaller its area. We will assign an area of 1.0 to the concepts corresponding to the *basic objects* of the hierarchy in the sense described in Rosch *et. al.* [2]. For instance, some areas used in our experiments are:

```
area([wolf]) = 1.0
area(["Hector"]) = 0.1
area([animal]) = 3.0
area([thing]) = 10.0
```

In the following, we introduce formulas for the calculation of the energy level of specific concepts in overlays. To simplify the formulas we will define a *trimming* function as follows:

$$trim(x,y) = \begin{cases} 0 & \text{if} \quad x < 0 \\ x & \text{if} \quad x \in [0,y] \\ y & \text{if} \quad x > y \end{cases} \tag{1}$$

where we will omit y when its value is 1.

Overlays are built iteratively, by adding one concept at a time, starting with an empty overlay. We start by defining the *explicit energy* of concept c in overlay C as $een(C,c) < area(c)$. In the empty overlay, the explicit energy of all concepts is zero, for non-empty overlays, the explicit energy is determined recursively by the formulas defining the addition operation.

We will define the explicit energy of the overlay as the sum of the explicit energies of all the concepts in the overlay:

$$een(C) = \sum_c een(C,c) \tag{2}$$

We define two addition operations, differentiated by whether they consider or not the impact of the concepts.

The *direct addition* $\oplus^{NI}$ or simply $\oplus$ is a simple summation of the explicit energy, limited by the area of the concepts, and is defined through the following formulas:

$$\begin{aligned} & C' = C \oplus \{c, e\} \Rightarrow \\ & \quad een(C',c) = trim(een(C,c) + e, area(c)) \\ & \quad \forall c_x \neq c \;\; een(C',c_x) = een(C,c_x) \end{aligned} \tag{3}$$

The second, *impacted addition* operation $\oplus^I$ also considers the *impact* between concepts. Adding a concept c_1 with energy e through this operation automatically triggers the addition of a number of other concepts, with an energy proportional with e, defined by the value $impact(c_1, c_2)$ which has the dimensionality of a positive or negative ratio. The impact is not necessarily symmetrical.

The impacted addition operation $\oplus^I$ is defined through the formulas:

$$\begin{aligned} & C' = C \oplus^I \{c, e\} \Rightarrow \\ & \quad een(C',c) = trim(een(C,c) + e, area(c)) \\ & \quad \forall c_x \neq c \;\; een(C',c_x) = \\ & \qquad trim(een(C,c_x) + e \cdot impact(c,c_x), area(c_x)) \end{aligned} \tag{4}$$

The explicit energy of a concept in an overlay is the energy we explicitly added either through direct addition or through impact.

Finally we define the *recursive impacted addition* $\oplus^{RI}$ in such a way that the concepts added through impact would also create their own impacts.

Another way through which a concept can have energy in an overlay is through *overlapping* a concept which has explicit energy. The overlap between two concepts c_i and c_j is defined in the dimensionality of the area $overlap(c_i, c_j) \in \mathbb{R}^+$. The overlap is always smaller than the area of either concept $area(c_i) \geq overlap(c_i, c_j)$ and it is symmetrical $\forall i \forall j \; overlap(c_i, c_j) = overlap(c_j, c_i)$. If two concept's areas are identical and their overlap is identical with the area, the two concepts are indistinguishable, and thus are considered equivalent:

$$area(c_1) = area(c_2) = overlap(c_1, c_2) \overset{def}{\Longrightarrow} c_1 \equiv c_2 \tag{5}$$

We define the *energy* of concept c in overlay C conservatively by:

$$\begin{aligned} & en(C,c) = \\ & \quad \max\Bigg(area(c), een(C,c) + \\ & \quad \max_{c_x \neq c}\Big(overlap(c_x, c) \cdot \frac{een(C,c_x)}{area(c_x)}\Big)\Bigg) \end{aligned} \tag{6}$$

The advantage of this conservative metric (i.e. relying on a max rather than a sum) is that the areas shared by three or more concepts do not need to be considered, as they do not enter into the calculation of the energy levels of the individual concepts in an overlay.

For a certain overlay C and concept c we can define the *activation* $act(C,c) \in [0,1]$ of the concept in the overlay with:

$$act(C,c) = \frac{en(C,c)}{area(c)} \tag{7}$$

Having defined the formulas governing concepts and COs, let us investigate the intuitions behind them. The definition of a concept and related terms in Xapagy does *not* map to scientific classification, computational logic, descriptive logic, or the possible worlds interpretation. The Xapagy system, in its current version, can not model these abstractions[1].

[1] If such abstractions must become part of the narrative, they need to be modeled in the story itself.

Roughly, Xapagy concepts cover the categories of nouns *and* adjectives of classical grammar. There is no notion of a "class" or "type" in Xapagy: there are only instances which happen to share certain attributes. A man is simply a random identifier which happened to have the attribute `[man]`.

The area of a concept, as we said is a metric of its specificity - more general concepts have a larger area. However, we should caution the reader against pushing the analogy between this metric and the area of two-dimensional shapes too far. For instance, we can not assume that the sum of the areas of the different animal types will be the area of the concept `[animal]`.

In a similar vein, Xapagy verbs cover both the categories of verbs *and* adverbs of classical grammar. A certain verb word in a sentence maps to an overlay of attributes. An interesting future research direction could be to investigate the relationship between this composition model and Pinker's *microfeatures* [1].

B. Verbs and verb overlays

The Xapagy system treats verbs very similarly to concepts. A *verb* or *verb concept* in Xapagy is the representation of an *undivisible attribute* of an action or event. A weighted superposition of verbs is called a *verb overlay* (VO). When talking about verbs in general, we will denote them with $v_1, v_2 \ldots$ and VOs as $VC_1, VC_2 \ldots$. For specific verbs we will use descriptive names in brackets such as `[hits]`. The formulas introduced for COs will hold for VOs as well.

In contrast to concepts which are fully defined by their overlays and impacts, Xapagy defines a group of special *meta-verbs* which, when inserted into the focus, trigger *side-effects*, in the form of modifying the focus and creating relations between instances and verb instances.

C. Negation

For each concept c we automatically define its negation $-c$. When we write out the name of the concept, such as in `[alive]`, we shall write the negation as `[not-alive]`. The negated concept is defined by its specific impact and overlap with reference to other concepts.

$$\begin{aligned} impact(c_1, -c_1) &= impact(-c_1, c_1) = -1 \\ overlap(c_1, c_2) &= overlap(-c_1, -c_2) \\ impact(c_1, c_2) &= impact(-c_1, -c_2) \end{aligned} \tag{8}$$

The negation defined in Xapagy does not follow logical (or arithmetical) rules of negation. The concept `[alive]` does not stand for all the instances which are alive, nor `[not-alive]` stay for all the other instances.

The definition of the negation, however, is designed to serve the needs of narrative reasoning, and most of the times yields results consistent with the commonsense interpretation:

```
"Hector" / is-a / warrior.
He / is-a / alive.
"Hector" / changes / not-alive.
```

The new instance of Hector will have the attributes `[Hector, not-alive]`, the `[alive]` attribute being removed by the negative impact of the `[not-alive]` concept[2].

III. Binary relations between overlays

A. Incompatibility metrics

One of the primary metrics between two overlays is the level of their incompatibility. Intuitively, incompatibility appears if the two overlays can not be logically merged into a common overlay. The incompatibility metric has extensive applications in the functionality of the Xapagy system. For instance, when resolving references, we can ignore items which have overlays incompatible with the reference. Incompatibility is also a negative factor for the shadows. In headless shadows an FSL can not support an incompatible interpretation.

[2]We can, of course, define a Xapi word "dead" which maps to `[not-alive]`.

To discuss the incompatibility we will always refer to two overlays: the source overlay `ovrSrc` and the reference overlay `ovrRef`.

The metric of incompatibility is assembled from two subcomponents:

- **negative impact energy:** merging the `ovrRef` to the `ovrSrc` would result in negative impacts removing some of the components of `ovrSrc`. This happens if `ovrRef` and `ovrSrc` contain a concept and its negation respectively (either directly, or through overlaps).
- **over-impact energy:** merging the `ovrRef` to `ovrSrc` would yield in some concept being over-reached in the impact. This can happen, for instance if two different concepts from the same category are present in `ovrRef` and `ovrSrc`.

Let us now see how the incompatibility metrics are calculated.

The incompatibility metrics allow us to pass a source object which might not have all the impacts resolved. The first step is to resolve the impacts in the overlay (this can be done by adding it in a resolved impact manner to an empty overlay).

`ovrSrcResolved` = `{}` $\oplus_I$ `ovrSrc`

`ovrRefResolved` = `{}` $\oplus_I$ `ovrRef`

The second step is to calculate the surplus of the reference over the source:

`ovrRefSurplus` = `ovrRef` $\ominus$ `ovrSrcResolved`

Let us now consider the application of the impact function as follows:

{positive impact, negative impact, overimpact} = `ovrSrcResolved` impact `ovrRefSurplus`

The negative impacts are the amount of energy which is removed by the impacts of `ovrRefSurplus`. For instance if for a concept of size 2.0 `c` is in `ovrSrcResolved` and `not-c` is in `ovrRefSurplus`, then the negative impact will accumulate a value of 2.0. The negative impact is not offset by the positive impacts, only calculated separately. The maximum value of the negative impact is the energy of `ovrRefResolved`.

The overimpacts are the amount of energy with which the original value and the value added by the impact exceed the size of a concept. Let us assume that we have a concept `c2` of size 4.0. This concept is present with energy 3.0 in the `ovrSrcResolved` and is impacted with energy 3.0 from `ovrRefSurplus`. Thus, this will appear with value 2.0 in the overimpact. Again, the maximum value of the overimpact is the energy of `ovrRefResolved`.

Another interesting observation is that the sum of the negative impact and overimpact is also bounded by the energy of `ovrRefResolved` because one specific energy chunck can be either part of a negative impact or overimpact, but not both.

To simplify processing in other parts of Xapagy, it is easier to work with a single metric with a known range. Thus we will integrate the two components into a single metric by defining the incompatibility score as follows:

$$\begin{aligned} &\texttt{incompatibility} = \\ &\qquad \frac{\texttt{negative impact} + \texttt{overimpact}}{area(\texttt{ovrRefResolved})} \end{aligned} \tag{9}$$

This value will always be in the range of 0 to 1.0.

B. Coverage metric

The coverage metric between two overlays measures the degree at which the attributes of a *test overlay* can be found in the *base overlay*. The metric returns 0 is the two overlays are completely disjoint, and 1 if the test overlay is completely contained in the base overlay. The metric is not symmetrical.

To discuss the coverage we will always refer to two overlays: the source overlay `ovrSrc` and the reference overlay `ovrRef`.

$$\texttt{coverage} = \frac{overlap(\texttt{ovrRef}, \texttt{ovrSrc})}{totalenergy(\texttt{ovrRef})} \tag{10}$$

The overlap component looks for all the concepts in the `ovrRef` and checks at what energy levels are present in `ovrSrc` - either directly or indirectly. Then, it adds it together. This sum can be at most the value of the energy in `ovrRef`. The coverage value divides this with the total energy of `ovrRef`, thus reaching a value between 0 and 1.0.

C. Resolution confidence metric

The resolution confidence metric measures the confidence at which we can say that a certain overlay refers to another overlay. This is a direct implementation of the resolution affordance. Whenever we are referring to an instance from a group (typically from the focus), the resolution confidence metric allows us to select the referred entity. Conventionally, we will calibrate this metric in such a way that a negative value means that the reference is not possible, while higher values make references increasingly likely.

In practice, in order to determine if an overlay C_{ref} refers to which of the instances $I_1, I_2 \ldots I_n$ we calculate the resolution confidence pairs $resconf(C_{ref}, (I_i), b_i)$ where b_i is a bias factor. The resolution will be chosen to be the instance with the largest resolution confidence, as long as it is larger than zero.

Let us now choose a resolution confidence which matches our intuition. The resolution confidence increases with the coverage of level of the reference, and it decreases with the incompatibility. Finally, for close cases, we can assume that we will have some pre-existing biases towards certain resolutions. This might be a proximity bias - for instance, assuming that recently referred instances will be referred again - or it can be a bias towards interpretations which lead to more expected interpretations.

The current model used in Xapagy is a simple model of linear combination:

$$\texttt{resConf} = \texttt{bias} + 10 \cdot \texttt{coverage} - 100 \cdot \texttt{incompatibility} \tag{11}$$

The bias value used is the focus strength of the instance. As all the values of bias, coverage and incompatibility are in the range [0, 1], the possible range of resolution confidence is in [-100, 10]. As we see, this model establishes a strong relative importance of the metrics. For instance, a comparatively small incompatibility will disqualify the referred object. On the other hand, bias will become a deciding factor only in cases where the coverage and incompatibility are almost completely identical.

IV. Experiments and discussion

In the following we illustrate the operation of the overlay relations through a series of examples of overlays built using the artificial domain. We start with a examples involving concepts in Table I. In the following we discuss the relevance of these examples, based on their count in the table. For all the examples, the resolution confidence was calculated with a zero bias.

1. `ovrSrc=[c_bai1]`, `ovrRef=[c_bai1]`: this example shows the case of two overlays having a single concept of independent base attribute (BAI) type. There is no negative impact or overimpact. The incompatibility score is 0.0, as these two overlays are perfectly compatible. The source completely covers the reference so the value is 1.0. The resolution confidence, for this pair of values is 10, the maximum obtainable value (essentially, this is the case when we are referring with the exact value).
2. `ovrSrc=[c_bai1 c_bai2 c_bai3 c_bai4]`, `ovrRef=[c_bai1]`: this is a case when we are referring to a CO with one of its sub-COs. There is no negative or overimpact, the incompatibility score is 0. The coverage is 1.0 (as the source completely covers the reference). The resolution confidence is its maximum value 10.0. Notice that in this case the values are exactly the same as in the identity relation - the fact that the source has attributes not present in the reference are not penalized in the resolution confidence.
3. `ovrSrc=[c_bai1]`,`ovrRef=[c_bai1 c_bai2 c_bai3 c_bai4]`: this is the opposite case of the previous, where the reference is larger than the source. Again, there is no incompatibility between the values. However, the coverage level of the reference by the source is only 0.25. This still means a relatively large resolution confidence, but it will still loose out to references where the reference does not have additional attributes.
4. `ovrSrc=[c_bai1]`, `ovrRef=[not-c_bai1]`: one of the overlays have a single BAI while the other one its negation. In this case the negative impact energy is 1.0, the size of the concept, and the incompatibility score is 1.0. There is no overimpact. The coverage is 0.0. The resolution confidence is the smallest possible (-100.0).
5. `ovrSrc=["PN1"]`, `ovrRef=[not-"PN1"]`: this differs from the previous example through the fact instead of the base attributes with area 1.0, we have proper names which have a smaller area (0.1). The negative impact energy, accordingly is smaller, however, we will have the same incompatibility and coverage scores, and thus the same -100.0 resolution confidence.
6. `ovrSrc=[c_bai1 c_bai2 c_bai3 c_bai4]`, `ovrRef=[not-c_bai1]`: here we have an example where we have a clear negation conflict for the single attribute in the reference. Although the source overhead has other attributes, the resolution confidence will have full negative value. Intuitively, "the only attribute it has is wrong".
7. `ovrSrc=[c_bai1]`, `ovrRef=[not-c_bai1 c_bai2 c_bai3 c_bai4]`: here we have a simple source, and a relatively rich reference of four concepts out of which one it a negation conflict while the other ones are not present in the source. Interestingly, this value, has a lower overall conflict than then previous one (0.).
8. `ovrSrc=[c_bai1 c_bai2 c_bai3 c_bai4]`, `ovrRef=[not-c_bai1 c_bai2 c_bai3 c_bai4]`. In this case we have a rich source and reference, where 1 item is a negation, but the other ones are matches. The result is 0.25 in incompatibility, 0.75 in coverage, yielding a -17.5 on the resolution metric. This is one of those examples where a single negation conflict overpowers the resolution metric.

EXPERIMENTAL:

Some of the values do not quite cover what we would like to see here. Note that it our intuitions might not necessarily be the right be here.

For instance, 7. has a much lower conflict and actually higher resolution confidence than 6., although intuitively they should be about the same (or even 7. to have a lower resolution confidence). One way to solve this would be to eliminate the extra items from the

TABLE I
EXAMPLES OF INCOMPATIBILITY, COVERAGE AND RESOLUTION CONFIDENCE VALUES FOR CONCEPT OVERLAYS

No.	coSrc	coRef	NIE	OIE	incomp	cover	resconf
1	[c_bai1]	[c_bai1]	0.000	0.000	0.000	1.000	10.000
2	[c_bai1 c_bai2 c_bai3 c_bai4]	[c_bai1]	0.000	0.000	0.000	1.000	10.000
3	[c_bai1]	[c_bai1 c_bai2 c_bai3 c_bai4]	0.000	0.000	0.000	0.250	2.500
4	[c_bai1]	[not-c_bai1]	1.000	0.000	1.000	0.000	-100.000
5	["PN1"=0.100/0.100]	[not-"PN1"=0.100/0.100]	0.100	0.000	1.000	0.000	-100.000
6	[c_bai1 c_bai2 c_bai3 c_bai4]	[not-c_bai1]	1.000	0.000	1.000	0.000	-100.000
7	[c_bai1]	[c_bai2 c_bai3 c_bai4 not-c_bai1]	1.000	0.000	0.250	0.000	-25.000
8	[c_bai1 c_bai2 c_bai3 c_bai4]	[c_bai2 c_bai3 c_bai4 not-c_bai1]	1.000	0.000	0.250	0.750	-17.500
9	["PN1"=0.100/0.100 c_bai1 c_bai2 c_bai3 c_bai4]	[c_bai1 c_bai2 c_bai3 c_bai4 not-"PN1"=0.100/0.100]	0.100	0.000	0.024	0.976	7.317
10	["PN1"=0.100/0.100 c_bai1 c_bai2 c_bai3 c_bai4]	["PN2"=0.100/0.100 c_bai1 c_bai2 c_bai3 c_bai4]	0.000	1.000	0.196	0.976	-9.852
11	[c_bai1]	[c_bai2]	0.000	0.000	0.000	0.000	0.000
12	[c_bao3_1]	[c_bao3_2]	0.000	0.000	0.000	0.300	3.000
13	[c_bai1]	[c_cat0_cmi1]	0.000	0.000	0.000	0.000	0.000
14	[c_cat0_cmi1]	[c_cat0_cmi2]	0.000	1.000	0.500	0.000	-50.000
15	[c_catv3_cmiv1]	[c_catv3_cmiv2]	0.000	0.300	0.231	0.000	-23.077
16	[c_catv20_cmiv1]	[c_catv20_cmiv2]	0.000	2.000	0.667	0.000	-66.667
17	[c_bai2 c_bai3 c_bai4 c_cat0_cmi1]	[c_cat0_cmi2]	0.000	1.000	0.500	0.000	-50.000
18	["PN1"=0.100/0.100]	["PN2"=0.100/0.100]	0.000	1.000	0.909	0.000	-90.909
19	[c_cat0_cmi1]	[c_bai2 c_bai3 c_bai4 c_cat0_cmi2]	0.000	1.000	0.200	0.000	-20.000
20	[c_bai1 c_cat0_cmi1]	[c_bai1 c_cat0_cmi2]	0.000	1.000	0.333	0.500	-28.333
21	[c_cot3_cmo1]	[c_cot3_cmo2]	0.000	0.700	0.350	0.300	-32.000
22	[c_bai1 c_cot3_cmo1]	[c_bai1 c_cot3_cmo2]	0.000	0.700	0.233	0.650	-16.833

reference before we start the conflict resolutions.

END OF EXPERIMENTAL

Next, we discuss similar examples for verb overlays, as shown in Table II. In the following we discuss the relevance of these examples, based on their count in the table.

REFERENCES

[1] S. Pinker. *Learnability and cognition: the acquisition of argument structure*. Cambridge, Mass.: MIT Press, 1989.

[2] E. Rosch, C. Mervis, W. Gray, D. Johnson, and P. Boyes-Braem. Basic objects in natural categories. *Cognitive psychology*, 8(3):382–439, 1976.

TABLE II
EXAMPLES OF INCOMPATIBILITY, COVERAGE AND RESOLUTION CONFIDENCE VALUES FOR VERB OVERLAYS

No.	ovrSrc	coRef	NIE	OIE	scINC	scCOV	ResConf
1	[v_av0]	[v_av0]	0.000	0.000	0.000	1.000	10.000
2	[v_av0]	[v_av1]	0.000	0.000	0.000	0.000	0.000
3	[v_av0]	[not-v_av0]	1.000	0.000	1.000	0.000	-100.000
4	[v_av0 vm_ActionMarker vm_Successor=0.500/1.000]	[v_av0 vm_ActionMarker vm_Successor=0.500/1.000]	0.000	0.000	0.000	1.000	10.000