Text Completion using a Context-Integrating Dependency Parser

Amr Rekaby Salama *
Department of Informatics
Universität Hamburg
Germany

Özge Alaçam *
Department of Informatics
Universität Hamburg
Germany

Wolfgang Menzel
Department of Informatics
Universität Hamburg
Germany

{salama,alacam,menzel}@informatik.uni-hamburg.de

Abstract

Incomplete linguistic input, i.e. due to a noisy environment, is one of the challenges that a successful communication system has to deal with. In this paper, we study text completion with a data set composed of sentences with gaps where a successful completion cannot be achieved through a uni-modal (language-based) approach. We present a solution based on a context-integrating dependency parser incorporating an additional non-linguistic modality. An incompleteness in one channel is compensated by information from another one and the parser learns the association between the two modalities from a multiple level knowledge representation. We examined several model variations by adjusting the degree of influence of different modalities in the decision making on possible filler words and their exact reference to a non-linguistic context element. Our model is able to fill the gap with 95.4% word and 95.2% exact reference accuracy hence the successful prediction can be achieved not only on the word level (such as *mug*) but also with respect to the correct identification of its context reference (such as *mug_2* among several mug instances).

1 Introduction

Text completion/prediction is a crucial element of communication systems, due to its role in increasing the fluency and the effectiveness of the communication in scenarios where the environment is noisy, or the communication partner suffers from a motor, or cognitive impairment (Garay-Vitoria and Abascal, 2004). In this study, we tackle the problem of compensating the incompleteness of the verbal channel by additional information from visual modality. This capability for multi-modal integration can be a very specific yet crucial feature in resolving references and/or performing commands for i.e. a helper robot that aids people in their daily activities. To the authors' knowledge, there is no multi-modal data set for a text completion task that systematically addresses challenging linguistic structures (i.e. syntactic or referential ambiguities) for environments where helper robots, who have access to visual information, would be employed.

The completion is performed by predicting tenable fillers for the missing, unknown, or vague parts in the input sentences through varying techniques, using single or hybrid methods. The prediction process utilizes the available resources, usually linguistic information (morphological, syntactic, and semantic properties). It can also use additional information sources such as the linguistic, or visual context (Garay-Vitoria and Abascal, 2006). If only the linguistic level is available, a language model can be used to predict the probability of a syntactic category in a certain context (Asnani et al., 2015; Bickel et al., 2005). N-grams is a popular method for this task since they provide very robust predictions for local dependencies. Nevertheless, they loose their power for structures with long-range dependencies. Furthermore, if there are multiple instances of the same object class (c.f. Figure 1), a text completion based on N-gram could not differentiate between them to select the proper instance reference. As shown in several studies (Mirowski and Vlachos, 2015; Gubbins and Vlachos, 2013), a language model employing the syntactic dependencies of a sentence brings the relevant contexts

*These authors contributed equally to this work

Proceedings of the 3rd Workshop on Representation Learning for NLP, pages 41–49
Melbourne, Australia, July 20, 2018. ©2018 Association for Computational Linguistics

closer. Using the Microsoft Research Sentence Completion Challenge (Zweig and Burges, 2012), Gubbins and Vlachos (2013) have showed that incorporating syntactic information leads to grammatically better options for a semantic text completion task.

On the other hand, semantic clustering or classification (like in ontologies) can be used to derive predictions on the semantic level. However, when it comes to the description of daily activities, contextual information coming from another modality would be more beneficial, since linguistic distributions alone could hardly contribute enough clues to distinguish the action of *washing a pan* from *washing a mug*, which is a crucial difference for helper robots.

A popular trick in natural language processing consists in training a model on one task, and then apply it to an entirely different one. We adopt this method by training a multi-modal dependency parser using noise-free sentences combined with a description of their visual context. In the second step, we make use of the trained parser to predict the best fillers of the gaps (guided by the context modality).

The paper starts by introducing our multi-modal approach for the text completion task. In section 3, we present the experimental setup including the compiled dataset. The implementation is described in section 4. Experimental results are presented and discussed in section 5. Conclusions are drawn and future directions of research are pointed out at the end of the paper.

2 A Multi-Modal Approach for a Text Completion Task

Although closing the gaps in a sentence based only on a language model is a simple way to tackle the issue, in extremely ambiguous situations, gap reconstruction is almost impossible on a purely unimodal base. In this paper, we work on multi-modal data that consists of linguistic and context information. The linguistic part is provided by natural language sentences that refer to a particular visual scene. The context information is a meta-data description of that scene. Per input sentence, the context channel contains a set of context relations: $(argument, relation\_type, predicate)$ where $relation\_type$ is one of a predefined set of accepted relations, such as agent or theme while $Predicate$ and $Argument$ are tokens of the input sentence. The complexity of the text completion task is controlled by creating challenging scenes along the following dimensions:

- Each scene is composed of different components (i.e., persons and objects).
- A scene might contain multiple instances of the same class (i.e., a blue mug (id: *mug_1*) and a green mug (id: *mug_2*).
- The different instances are taking part in various relations (more details are given in Section 2.2).

In a series of experiments, we assess the potential of a context-integrating dependency parser for correctly solving the text completion task. We not only try to determine whether we can fill the gap in the sentence with the correct word but also whether it is possible to correctly determine the exact reference to an entity in the context description given the contextual information, in particular if the linguistic input is noisy and a token of the input sentence is missing. At this stage of research, we work only on one gap per sentence.

2.1 Context-integrating Dependency Parser

Dependency parsing is an essential NLP task that determines the syntactic structure of the input sentence in form of a dependency tree. Each token of the input is represented as a tree node. The tree consists of the dependency relations between each word of the sentence and its *head* word (Nivre, 2004).

The standard input of a parser is a natural language sentence. To supply such a parser with additional information required for text completion in a multi-modal environment we have to make it sensitive to cues from the context.

In our previous research (Salama and Menzel, 2018, 2017), we have introduced a multi-modal dependency parser adopting the graph-based approach of Eisner (1996) and Mcdonald and Pereira (2006). Our model, called RBG-2, extends the RBG parser (Zhang et al., 2014) by enabling multi-channel input providing the parsing process with context information in addition to the natural language sentence. Integration is achieved by combining features from both input channels during the normal training procedure of the RBG parser.

2.2 The Data Set

In order to test how the model behaves for different linguistic structures, we used the nine different grammatical templates[1] given in Table 1 featuring active/passive voice, PP-attachments, relative clause (RC) attachments, and conjunctions. They are combined with several actions performed by different agents. The dependency structures are represented in the CONLL-X format. The data set consists of 429 individual sentences for 20 different visual scenes. We performed a 10-fold cross validation and introduced exactly one gap for either a noun, verb or adjective into each test sentence obtaining 1457 test sentences in total.

2.2.1 Linguistic Structures

In this section, we examplify the nine grammtical templates used in our data-set. The following examples belong to the scene in Figure 1:

- **T1. RC[2] Attachment Ambiguity-1**
  T1A. *Active voice in RC.* "It is a mug on a vitrine that the woman damages."
  Either the relative clause is low-attached (*the woman damages the vitrine*) or high-attached (*the woman damages the mug*).
  T1B. *Passive voice in RC.*"It is a mug on a vitrine that is damaged by the woman."
- **T2. RC Attachment Ambiguity-2**
  T2A. *Active voice.* "The woman damages the vitrine with a mug on it."
  T2B. *Passive voice.* "The vitrine with a mug on it is damaged by the woman."
- **T3. RC Attachment Ambiguity with a Genitive Object-3**
  T3A. *Active voice in RC.*"The woman removes the label of the medicine that lies on the shelf "
  T3B. *Passive voice in RC.* "The label of the medicine that lies on the shelf is removed by the woman."
- **T4. Scope Ambiguity**
  "There are a mug, a candle and books [that lie/lying] on the vitrine."
- **T5. Simple Imperative sentence**
  "bring me the mug [that lies/lying] on the vitrine [that the woman cleans]."
- **T6. Imperative sentence with modifiers**
  "bring me the blue mug [that lies/lying] on the vitrine [that the woman cleans]."

2.2.2 Context Representations

The visual information of a picture is represented in a knowledge base that contains the relationships between objects, characters and actions in the scene. This information has been manually annotated as triplets composed of argument, relation type and predicate. Currently, we consider six different context relations, namely *agent*, *theme*, *location*, *next-to*, *part-of/own*, as well as *property* assignments for color, material, shape etc. (e.g., *a blue mug* or *a ceramic vase*). Figure 1 exemplifies the context annotations of a visual scene with an additional concept map representation (bottom). In this scene, *the woman* is the *agent*, who performs the cleaning action, the *vitrine* is the *theme*, i.e. the entity undergoing a change of state, caused by the action. The entire data set and source code can be accessed from `https://github.com/rekaby/MD-TC.V1.0`

For the current study, the pictures, as the one given in Figure 1, serve illustrative purposes, because the computational model does only have access to the manually annotated representations. An automatic relation extraction is not within the scope of this study.

The different semantic roles are distributed in the data set as follows; Agent (%13.6), Theme (%13.6), Location (%33.1), Next to (%9.8), Property (%19.5), Own (%10.3). Table 2 presents a statistics for the amount of contextual information per scene.

3 Implementation

RBG-2 parser starts by creating a fully connected graph representing the input tokens as nodes. The parser decodes a minimum spanning tree out of the graph maximizing the aggregated scores of the arcs. The scores are calculated by combining the weights of linguistic features and context features between the pair of tokens as follows:

$$\overline{y} = \max_{y \in T(x,c)} \sum_{i=1}^{n} \omega_l . f(x_i, x_j, y) + \omega_c . \widehat{f}(c_i, c_j, y) \tag{1}$$

Where $\overline{y}$ is the best dependency tree, $T(x,c)$ is a set of all possible dependency trees for input sentence X and context c. The linguistic feature vector between node x_i and its dependency head x_j is

[1] inspired by the experimental setup of pyscholinguistic research

[2] Relative Clause

Types	Templates	Sentences	Gaps	Gap Types
T1A	PRO1$_{\text{nom}}$ VP1 NP1$_{\text{acc}}$ NP2$_{\text{gen}}$, WDT*$_{\text{acc}}$ PRO2$_{\text{nom}}$ VP2	41	130	NP(114), VP(16)
T1B	PRO1$_{\text{nom}}$ VP1 NP1$_{\text{acc}}$ NP2$_{\text{gen}}$, WDT*$_{\text{acc}}$ PRO2$_{\text{nom}}$ VP2	40	130	NP(115), VP(15)
T2A	PRO$_{\text{nom}}$ VP1 NP1$_{\text{nom,pl.}}$ NP1$_{\text{nom,pl.}}$, WDT $_{\text{acc,pl.}}$ VP2 PP1	50	177	NP(153), VP(24)
T2B	PRO$_{\text{nom}}$ VP1 NP1$_{\text{nom,pl.}}$ NP1$_{\text{nom,pl.}}$, WDT $_{\text{acc,pl.}}$ VP2 PP1	50	177	NP(153), VP(24)
T3A	NP$_{\text{it}-\text{cleft}}$ VP1 NP1$_{\text{nom}}$ NP2$_{\text{dat}}$, WDT $_{\text{dat}}$ PRO$_{\text{3rd-sing.}}$ ADV VP2	36	177	NP(139), VP(36), ADJ(2)
T3B	NP$_{\text{it}-\text{cleft}}$ VP1 NP1$_{\text{nom}}$ NP2$_{\text{dat}}$, WDT $_{\text{dat}}$ PRO$_{\text{3rd-sing.}}$ ADV VP2	29	145	NP(113), VP(30), ADJ(2)
T4	EX V$_{\text{aux}}$ NP1$_{\text{nom}}$ (CONJ NP2) WDT*$_{\text{nom}}$ VP1 Prep. NP3	58	156	NP(154), VP(0), ADJ(2)
T5	VP1 (Pro$_{\text{dat}}$ NP1 (WDT) VP2 Prep. NP2	63	141	NP(137), VP(1), ADJ(3)
T6	VP1 (Pro$_{\text{dat}}$ (Adj1) NP1 (WDT) VP2 Prep. (ADJ2) NP2	62	224	NP(135), VP(9), ADJ(80)
Total		429	1457	NP(1213), VP(155), ADJ(89)

Table 1: POS templates, the number of sentences and gaps for each sentence types, and the number of gaps for each POS category

Items	Mean
Relations (*Min.*=28, *Max.*=41)	34.8
Context entities (*Min.*=30, *Max.*=41)	35.6
Unique entities [5] (*Min.*=21, *Max.*=28)	25

Table 2: Complexity of the contextual information for the visual scenes in the data set

Algorithm 1 Text Completion Workflow using RBG-2

```
  TR-L ← Training data (complete sentences).
  TR-C ← Training data (context).
  TE-L ← Testing data(sentences with gaps).
  TE-C ← Testing data (context).
  model ← train RBG-2(TR-L, TR-C)
  for each pair TE-L$_i$,TE-C$_i$ do
      $bestFillerScore \leftarrow -Inf.$
      for each component TE-C$_{ij}$ $\in$ TE-C$_i$ do
          for each POS$_t$ $\in$ POS tags do
              TE-L$_i$ ← SetGap(TE-C$_{ij}$,POS$_t$).
              $score$ ← parse(model,TE-L$_i$)
              if $score > bestFillerScore$ then
                  $bestFillerScore \leftarrow score.$
                  $bestFiller$ ← TE-C$_{ij}$.
                  $bestPOS$ ← POS$_t$.
      TE-L$_i$ ← fillGap( bestFiller,bestPOS).
```

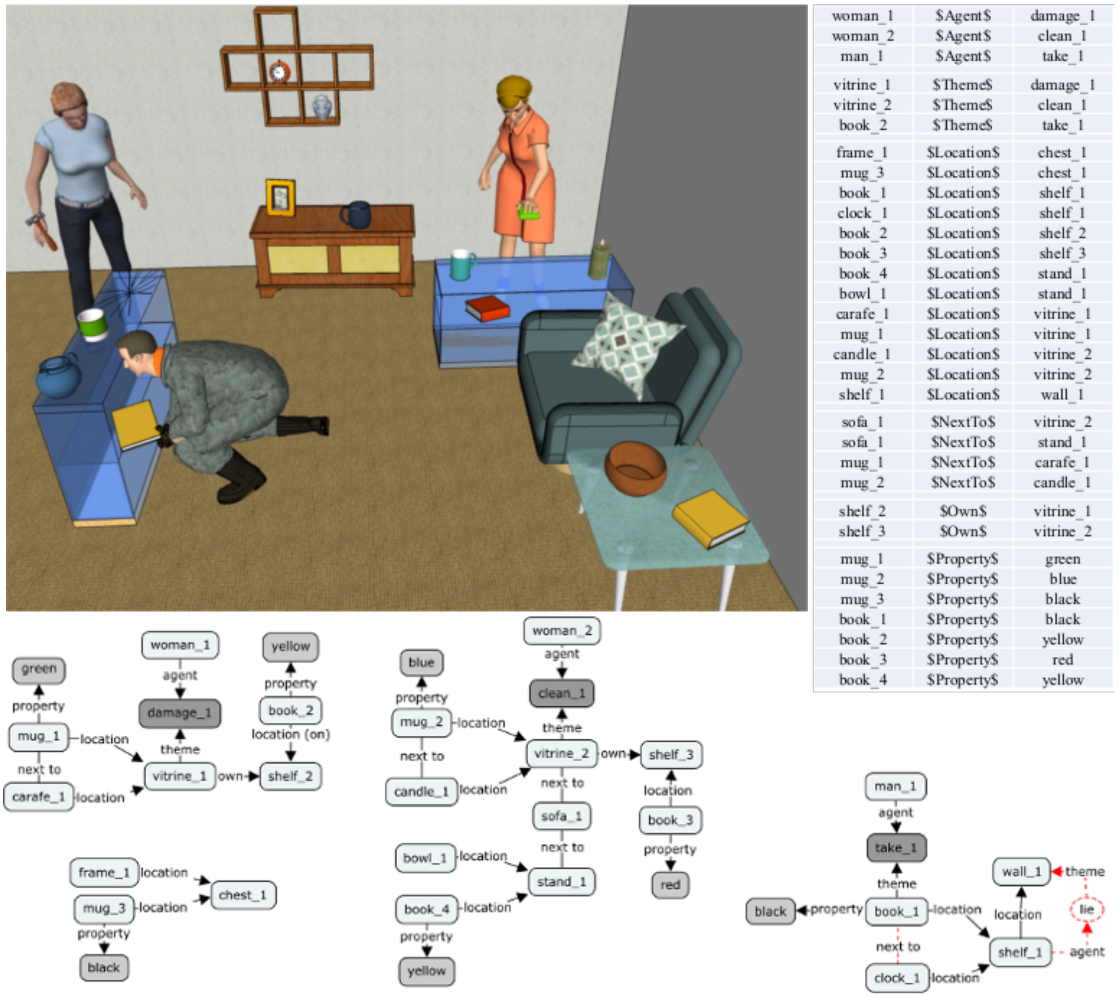
$f(x_i, x_j, y)$ with weights ω_l. The context feature vector is $\widehat{f}(c_i, c_j, y)$ with weights ω_c.

We build the context features using combinations of the predicates' and arguments' POS, lemma, and word. So far, we only use first-order features for both channels. That means, only information about immediately connected nodes in the graph (head-child relationships) is accounted for, but more complex, indirect connections (siblings, grantchilds, etc.) are ignored. We add the context features to the graph arc only if the pair of nodes (words) has a context relation in between. Using no higher-order features makes the learning process faster and simpler but introduces some limitations as discussed in the result section. Each record in the training data set consists of a complete input sentence, a set of context relations such as in Figure 1, and an exact context reference for each sentence token (if it exists in the context description).

In the testing (text completion) phase, the input sentence is incomplete (containing exactly one gap) while the context information is the same as in the training phase except the mapping between the input sentence and context references is missing as well. E.g. in the sentence *"The vitrine next to sofa is cleaned by the GAP"* accompanying the scene in Figure 1, we have multiple goals to determine

- the filler word *(woman)*,
- the context filler reference *(woman_2)*,
- the context filler reference for all the other non-gap tokens in the input (if they exist). They are {*vitrine_2, sofa_1, clean_1*},
- the POS tag of the filler *(NN)*.

As shown in Algorithm 1, we train our data-driven RGB-2 parser on the multi-modal training set described above to learn the associations between the context knowledge representation and the the dependency structures. In the testing phase, we fill the gap by all the possible context components and parse the sentence in a multi-modal setup. We also iterate over different POS tags for the filler to compare the resulting dependency tree scores. The best filler (word, context-reference, and POS) means that this word/context-reference



woman_1	$Agent$	damage_1
woman_2	$Agent$	clean_1
man_1	$Agent$	take_1
vitrine_1	$Theme$	damage_1
vitrine_2	$Theme$	clean_1
book_2	$Theme$	take_1
frame_1	$Location$	chest_1
mug_3	$Location$	chest_1
book_1	$Location$	shelf_1
clock_1	$Location$	shelf_1
book_2	$Location$	shelf_2
book_3	$Location$	shelf_3
book_4	$Location$	stand_1
bowl_1	$Location$	stand_1
carafe_1	$Location$	vitrine_1
mug_1	$Location$	vitrine_1
candle_1	$Location$	vitrine_2
mug_2	$Location$	vitrine_2
shelf_1	$Location$	wall_1
sofa_1	$NextTo$	vitrine_2
sofa_1	$NextTo$	stand_1
mug_1	$NextTo$	carafe_1
mug_2	$NextTo$	candle_1
shelf_2	Own	vitrine_1
shelf_3	Own	vitrine_2
mug_1	$Property$	green
mug_2	$Property$	blue
mug_3	$Property$	black
book_1	$Property$	black
book_2	$Property$	yellow
book_3	$Property$	red
book_4	$Property$	yellow

Figure 1: The corresponding image[4] for the sentences above and the semantic representations of the actions and relations in the image

is the best matching one that combines two perspectives: grammatical correctness and compatibility with the context information.

Although the ratio of contextual features to syntactic ones (first-order features) is 1:2.3, which is not high, trying all the possible context elements is rather expensive. For each sentence, we need to build $G * C * P * M$ dependency trees that have to be ranked to find the best one. Here, G is the number of gaps (1 in our experiments), C the number of context entities (35.6 in average), P the number of PoS tags (3) and $M = \prod_{i=1}^{N} M_i$, where M_i is the count of possible candidates references and N the number of sentence tokens with probable context references.

The search space could be reduced by avoiding irregular combinations of POS and filler words. In this stage of research, however, we do not prune it at all.

3.1 Context Data Preprocessing

In a preprocessing phase, we enrich the context information by inferring new relations from the original ones (colored red in Figure 1 and Figure 2). We have used two kinds of inferred relations:

Location to Agent/Theme: If we have a context relation such as $(X, Location, Y)$, this might appear in the linguistic modality in two different forms having either direct or indirect syntactic dependency. For example, mug and $vitrine$ as in Figure 2A and 2B have a direct syntactic dependency and context relation respectively. In other sentence forms as in Figure 2C and 2D, there is no direct correspondence between the linguistic dependency and the context relation. Contextually, the two tokens are related through the $Location$ relation, but syntactically they are daughters of the same action lie (no direct dependency). In this form, the $Location$ relation is presented in the lin-

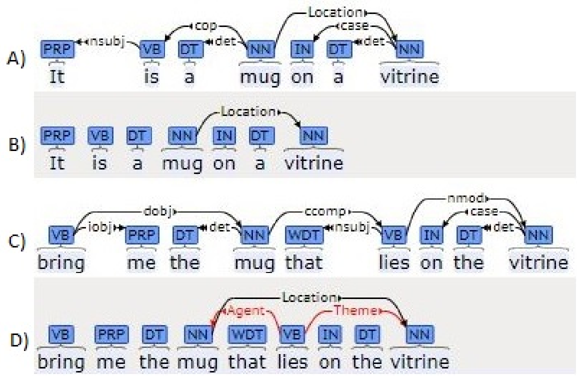


Figure 2: Location Inferred Agent/Theme Relations: A and C) dependency trees, B) original context information, D) original and inferred context relations

guistic modality using the verb to lie, which does not appear as a predicate in the context description.

To enrich the context repreresentation with information corresponding directly to the linguistic one, we define a set of verbs (LV) that have a location meaning (i.e., lie, stand, hang). From any location relation $(X, Location, Y)$, we infer another two relations $(X, Agent, LV_i)$ and $(X, Theme, LV_i)$, where $LV_i \in LV$ and LV_i is a token in the input sentence.

Location to Next_To relations: Given each pair of location relations $(X, Location, Z)$ and $(Y, Location, Z)$ we infer new relation $(X, Next-To, Y)$, where $X, Y, Z \in W$, W is the set of the input tokens. The inferred relations are added to the original list of the context input. In the rest of this paper, we use (IC) to refer to the Inferred Context relations and (OC) for the Original Context relations.

3.2 Model Variations

Varying syntactic/context's weight ratios (S2C): In the testing phase, we experiment with different ratios giving more influence (weight) to the context relations than to the linguistic ones. We assess different ratios (1to1,1to5, 1to10, and 1to25).

Original/Inferred relations' weight ratio (OC2IC): Similar to *S2C*, in the testing phase, we give more weight to the original relations than to the inferred ones by assigning the *OC2IC* ratio to 5to1.

4 Results

In order to show the effect of contextual information and to optimize the performance of the current model, we carried out several experiments with different parameters of the model by keeping the data set constant. We used 18 scenes (386 sentences in average) for training and kept the remaining 2 scenes (146 sentences on average) for test using a 10-fold cross-validation. In case the gap can be filled with more than one reference ($< 5\%$ of our dataset), we consider any possible one of them as correct. We used five evaluation metrics as listed below.

- POS-tag Accuracy
- Filler Word Accuracy
- Exact Filler Identification (*EFI*) Accuracy (i.e *mug_1* in contrast to *mug_2*)
- Non-gap Identification Accuracy, for all the other tokens in the input sentence.
- Complete Sentence Identification Accuracy
- Dependency Tree Accuracy (unlabeled attachment score, UAS)

Table 3 presents the results obtained from different variations of the model described in the previous section. We test a uni-modal parser (linguistic-only) only to show that the data set indeed is consisting of sentences, where reference resolution/text completion cannot be achieved on a purely uni-modal sense. For that purpose, the contribution of contextual information is turned off. Because of the uniform structure of the training dataset, the POS and dependency tree accuracies are very high 97.6% and 95.6% respectively. However, the model's prediction performance is drastically low for the gap words; 13.5% for the filler word and 7.8% for the exact filler identification.

As described in Section 3.2, the first model is based on having equal weights (S2C-1to1) for both syntactic (S) and contextual features (C) and the weights of original contextual (OC) features to the inferred features (IC) are kept equal as well (OC2IC-1to1). Giving equal weights leads to approx. 83% accuracy in both filler word and exact filler ID predictions, while increasing the influence of the context resulted in 95% accuracy[6].

[6] The other models with weights > 5 produced almost similar results.

Furthermore, giving more weight to the original relations over the inferred ones resulted in lower accuracy, therefore OCtoIC-1to1 variation is chosen as the standard for the analysis in this section. It is apparent from Table 3, a higher influence of the context is beneficial for a correct reference prediction. However, it should be noted that giving more weight to contextual features causes the model to be less sensitive about choosing a correct dependency tree. A closer look at the differences between the predictions of the S2C-1to5 and S2C-1to10 variations showed that 60 instances either in the dependency tree or in the filler ID were observed in the results. While S2C-1to5 builds 51 correct dependency trees and 43 correct references, S2C-1to10 chooses the correct dependency tree in only 12 instances, but even if the dependency tree is wrong, it fills the gap correctly in 48 out of 60 instances.

95 inaccurate EFI in 73 test sentences were observed. False predictions of the model variations can be categorized into several groups:

Inferred Relations. 60% of the inaccurate predictions occurred within this category. As explained in the Section 3.1, a phrase like *"an entity-1 that lies on an entity-2"* can be resolved due to an inferred relation. However, for sentences containing structures like *"an entity-1 that lie/stand/hang(s) next to an entity-2"* with a gap in a position of *entity-2*, the model prefers the most plausible filler that has a *location* relation (either original or inferred) with the *entity-1* instead of having a *next to* relation with it.

A Chain of Relations. This problem arises when for example there is a chain of location relations among the entities (7.4%), i.e. (*bird_1*, *Location*, *cage_1*), and (*cage_1*, *Location*, *chest_1*) with a description *"It is a cage on a chest that the man cleans"* with a gap in a *chest* position. While the S2C-1to5 model correctly fills the gap, S2C-1to10 chooses *bird* for the gap position. Assigning more weight to the context information leads to similar scores for the various entities of the chain, which may cause some wrong filler predictions.

Less represented PP associations. Syntactically, all prepositions *(with, of, on and next to)* have the same PoS tag but semantically they differ. While preposition *of* is associated with the *own* relation, and preposition *next to* with *next-to*, there are two prepositions which are related to the *location* relation; *on/in* and *with*. The distribution of them is as follows; *with*: 21.3%, and *on/in/under*[7]: 78.7%. As shown in Figure 3, the most likely association between syntactic and contextual features (w.r.t. *location* relations) is *head to argument* and *dependent to predicate*. This association is flipped for the prepositional phrase like *"entity-2 with entity-1 on it"*. Regardless of giving more influence to the context in that case, the model makes the prediction more strongly biased to the canonical direction of prepositional phrases resulting a wrong text prediction.

A Verb in a Noun Position. This error occurs irrespective of the linguistic structure if more weight (1to10 or 1to25) is given to the context (6.3%). As an example, a gap in the *shelf* position in the sentence *"There are a cat, a flower and books on the shelf"* is filled with *chase*, caused by the *(cat_1, Theme, chase_1)* relation. In that case, a stronger contextual influence overrides the syntactic form of the PP-attachment, and favors a reference with the *theme* relation, which has a consistent syntactic representation; its argument always points to the predicate. The goal to find syntactically correct PP-attachment is overruled by the more powerful features of the context relations, and so *chase* is selected considering that *a cat* is the only entity with a *theme* relation among others.

Far-Attachments. Far-attachments of the relative clauses or prepositional phrases are not that frequent as short-attachments, yet they are grammatically correct and occur in a data set. The results indicate that giving more influence to contextual information (S2C-1to10 and 1to25) helps to correctly fill the gap, while a model with lower weight for the contextual information (S2C-1to1 and -1to5) tends to choose the wrong reference for the gap position. To illustrate, the sentence *"It is a blanket on a couch that is grasped by the woman"* refers to one instance of a blanket class, and the context contains two instances: *blanket_1* and *blanket_2*, where *blanket_2* is the theme of the *grasp* action. When the gap is in *the couch* position, S2C-1to5 chooses a dependency tree with a short attachment of the RC. It attaches the gap to the action *grasp* and thus fills it with *blanket_2*. This is consistent with the *theme* relation in the context, resulting a sentence *"It is a blanket_1 on*

[7] excluding the occurrences of "*on/in/under*" in the reflexive phrases as in *"with a mug on it"*

Model Variations	PoS	Filler Word	EFI	Non-Gap	Complete Sentence	DP-UAS
Uni-model (linguistic only)	97.58	13.50	7.75	62.05	2.21	95.60
S2C Weight (1to1) + OC2IC Weight (1to1)	98.34	83.53	83.11	97.35	83.60	95.60
S2C Weight (1to5) + OC2IC Weight (1to1)	98.89	95.36	95.22	99.24	94.67	95.11
S2C Weight (1to10) + OC2IC Weight (1to1)	98.48	95.57	95.50	99.07	94.81	94.80
S2C Weight (1to25) + OC2IC Weight (1to1)	98.13	95.57	95.50	99.36	94.81	94.51
S2C Weight (1to5) + OC2IC Weight (5to1)	98.82	92.39	92.32	98.85	91.76	95.05

Table 3: The results of the different model variations

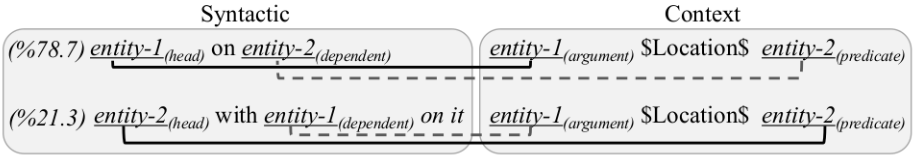

Syntactic
Context
(%78.7) entity-1(head) on entity-2(dependent)
entity-1(argument) $Location$ entity-2(predicate)
(%21.3) entity-2(head) with entity-1(dependent) on it
entity-1(argument) $Location$ entity-2(predicate)

Figure 3: Syntactic/Context feature association for the prepositions *on* and *with*

a blanket_2 that is grasped by the woman"[8]. If the context had only one blanket, that instance of the blanket had to be assigned to a non-gap blanket position in the sentence, and then the model is forced to switch to another dependency tree with a lower score but a better alignment. On the other hand, a 1to10 model gives more influence to the context, resulting in a correct completion even if the dependency structure is wrong. This may indicate that in order to deal with more challenging contexts or less represented linguistic structures (like far-attachments) increasing the influence of the contextual information would be beneficial.

Contextually Challenging Cases. This category covers 10.5% of the errors. To illustrate, lets consider a context, which contains two different roles for the same agent *man_1* together with a sentence like *"The handle of the mug on the counter is hold by the man"*; namely an action *wash* with a *theme* relation to a *mug* and another action *hold* with a *theme* relation to a *handle*. Another relevant relation for this sentence is *(mug_1, Own, handle_1)*. If in such a case the gap is in the verb position, the model can choose the alternative actions associated with a *mug* instead of forcing a far-attachment which is also favored contextually.

5 Conclusion and Future Directions

In this paper, we present a data set for sentence completion consisting of problematic instances, which can not be effectively handled using linguistic features alone. We apply a context-integrating dependency parser to solve this problem. There are number of assumptions and constraints of the current model. First, allowed gaps are only nouns, verbs and adjectives. Pronouns are not used as a possible gap filler. Furthermore, each individual instance is allowed to occur in the sentence once[9], thus a context reference (i.e. *mug_1*) can not be assigned to more than one token of the input sentence. Morever, the set of context relations is restricted to the the six relations. Further studies will need to cover more variety to relax these limitations.

The results indicate that incorporating contextual information and giving a strong enough influence to them helps to solve a majority of the problems concerning different sentence structures with conjunctions, relative clauses or PP-attachments. There are still some challenging situations originating from high degrees of linguistic or contextual complexity, which need to be addressed in future work. Furthermore, we plan to address noisier linguistic input with multiple gaps in a sentence as well as mismatches between the sentence and its contextual information. We also target reference resolution at the earliest time possible by employing incremental processing.

Acknowledgments

This research was partially funded by the German Research Foundation - DFG Transregio SFB 169: Cross-Modal Learning, and German Academic Exchange Service (DAAD).

[8] Our assumption is not using same context reference twice

[9] This constraint is not based on linguistic phenomena, it is just the design decision for the current solution

References

Kavita Asnani, Douglas Vaz, Tanay PrabhuDesai, Surabhi Borgikar, Megha Bisht, Sharvari Bhosale, and Nikhil Balaji. 2015. Sentence completion using text prediction systems. In *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*. Springer, pages 397–404.

Steffen Bickel, Peter Haider, and Tobias Scheffer. 2005. Learning to complete sentences. In *European Conference on Machine Learning*. Springer, pages 497–504.

Jason M. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '96, pages 340–345. https://doi.org/10.3115/992628.992688.

Nestor Garay-Vitoria and Julio Abascal. 2004. A comparison of prediction techniques to enhance the communication rate. In *ERCIM Workshop on User Interfaces for All*. Springer, pages 400–417.

Nestor Garay-Vitoria and Julio Abascal. 2006. Text prediction systems: a survey. *Universal Access in the Information Society* 4(3):188–203.

Joseph Gubbins and Andreas Vlachos. 2013. Dependency language models for sentence completion. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. pages 1405–1410.

Ryan Mcdonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *In Proc. of EACL*. pages 81–88.

Piotr Mirowski and Andreas Vlachos. 2015. Dependency recurrent neural language models for sentence completion. *arXiv preprint arXiv:1507.01193* .

Joakim Nivre. 2004. Incrementality in deterministic dependency parsing. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*. Association for Computational Linguistics, Stroudsburg, PA, USA, IncrementParsing '04, pages 50–57. http://dl.acm.org/citation.cfm?id=1613148.1613156.

Amr Rekaby Salama and Wolfgang Menzel. 2017. Multimodal graph-based dependency parsing of natural language. In *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016*. Springer International Publishing, pages 22–31.

Amr Rekaby Salama and Wolfgang Menzel. 2018. *Learning Context-Integration in a Dependency Parser for Natural Language*, Springer International Publishing, pages 545–569.

Yuan Zhang, Tao Lei, Regina Barzilay, Tommi Jaakkola, and Amir Globerson. 2014. Steps to excellence: Simple inference with refined scoring of dependency trees. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 197–207. http://www.aclweb.org/anthology/P14-1019.

Geoffrey Zweig and Chris JC Burges. 2012. A challenge set for advancing language modeling. In *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*. Association for Computational Linguistics, pages 29–36.