Dataset for extraction of assembly information from instruction manuals

Carlos M. Costa
Faculty of Engineering
University of Porto
Rua Dr. Roberto Frias, s/n 4200-465, Porto, Portugal
Email: carlos.costa@fe.up.pt

***Abstract*—Teaching industrial robots by demonstration can significantly decrease the re-purposing costs of assembly lines worldwide. To achieve this goal, the robot needs to semantically detect and track each object component with high accuracy. To speedup the object recognition phase, the learning system can gather information from assembly manuals in order to identify which objects are required for assembling the new product and if possible also extract the assembly order and spatial relation between them. This paper presents a dataset to test a Named Entity Recognition (NER) system with these goals. The dataset contains assembly operations for alternators, gearboxes and engines, which were written in a language discourse that ranges from professional to informal. For validating a given NER system with this dataset, each assembly operation has a list of the named entities and the required quantities for performing the product assembly. This allows to evaluate NER systems using precision, recall, accuracy and F1 scores. The dataset can also be used to evaluate information extraction and computer vision systems, since most assembly operations have pictures of the objects to assemble and also diagrams showing the necessary product parts, their assembly order and spacial disposition.**

***Index Terms*—Natural Language Processing Dataset, Named Entity Recognition, Small Parts Assembly, Industrial Robotics**

I. Introduction

Programming of industrial robots for assembly operations is a meticulous and arduous task that requires a significant engineering effort and long testing and deployment phases. For high volume manufacturing this cost is acceptable, but it is too expensive to repurpose robots for small volume production using traditional programming approaches. These issues can be overcome with robots that can learn new assembly skills by observing experienced operators and interacting with them through natural language. To achieve these goals, the robot needs to successfully recognize the objects within its workspace and semantically track their pose with high precision while the operator demonstrates how to perform the assembly operations. Moreover, it must be able to understand any instructions that the operator might give and also have the ability to recall them if asked later on. This type of teaching allows rapid reprogramming of flexible robotic assembly cells for new tasks. This process can be speed up further if there are assembly manuals available, which allows the robotic system to extract the objects and their assembly spacial disposition from the textual and visual representations. By knowing which objects to expect for a given teaching session, the object recognition efficiency can be significantly increased (by limiting the object search database). Moreover, this preliminary information extraction phase reduces the human teaching phase to only the operations that lack detailed information.

This paper presents a dataset for evaluating NER systems using assembly instructions of alternators, gearboxes and engines in several writing styles, from highly professional and structured text to colloquial and informal language. These assembly operations were extracted from Portable Document Format (PDF) files that besides textual descriptions also had assembly pictures and diagrams. As such, this dataset can be used for evaluating systems that combine both natural language processing algorithms and also computer vision and information extraction systems. For evaluating these Natural Language Processing (NLP) systems using this dataset, each assembly operation has a list with the main assembly objects and their required quantity for successfully performing the product assembly. For speeding up testing, it is provided two versions of the dataset, one with already tokenized text and another with the original text. Moreover, the dataset is already split into 75% of training text and 25% of testing text.

In the following section it will be given a brief overview of applications of NLP in robotics and also the main related work on extraction of assembly information from textual representations. Section III describes the main dataset sources for the 3 product types with assembly operations. Section IV presents the main steps that were performed to extract and clean the text from the PDFs. In Section V it is presented rank-frequency graphs for the tokenized dataset n-grams. Later on, it is shown the evaluation of the n-gram language models (from unigrams to pentagrams) and is also listed some sentences generated using these models. Finally, Section VI presents the conclusions and Section VII gives an overview of future work that can be done to extract assembly information from this dataset.

II. Related work

NLP algorithms have been integrated into robotics systems for a myriad of applications, ranging from control of industrial robotic arms [1], [2] and mobile robots [3] to complex interaction with humanoid systems using a combination of voice, text and image perception analysis [4], [5]. For the voice and textual teaching, the objects names and relations can be identified using NER algorithms [6], [7]. This type of approach usually relies on syntactic and semantic parsing of the text and also in machine learning algorithms [8] in order to be able to recognize previously unseen object names. It may present some challenges [9], but this methodology can achieve multilingual entity recognition [10] if language agnostic attributes are used.

Advanced applications of NLP algorithms include the teaching of assembly operations to robot arm manipulation systems by human operators. The JAST robot presented in [11] was implemented using a multi-agent system capable of learning assembly operations by interpreting human voice commands along with their gestures and gaze. The speech recognition system uses a Combinatory Categorial Grammar (CCG) and a semantics module to analyze if the operator is making statements for teaching, asking for information or giving answers to previous questions made by the robot. The vision system besides tracking the hands and gaze of the operator to perform a better speech analysis, it also recognizes the assembly objects within the robot workspace using template matching techniques.

Another example of usage of NLP methods in an industrial scenario is presented in [12], in which it is used a multi-lingual statistical semantic parser to extract assembly operations from natural language sentences given by remote operators. The system was developed using a client-server architecture, containing a natural language parser, a Knowledge Integration Framework (KIF) and an engineering system. The parser finds predicates with their respective arguments in sentences and establishes coreference chains. The KIF contains ontologies and semantically annotated skills that are used for filtering the predicates and return only the ones relevant for assembly operations. Lastly, the engineering system is a high level programming interface that uses the predicates found by the parser to select the appropriate skills for assembly while also matching the predicates arguments with the knowledge database in order to identify the objects and analyze in which branch of the assembly tree they must be inserted for achieving proper assembly order.

Besides voice and textual input from humans, the assembly information can also be retrieved from online web pages or knowledge repositories. The system proposed in [13] can extract the assembly graph by performing a syntactic and semantic analysis using a Probabilistic Context-Free Grammar (PCFG) parser and a Part-Of-Speech (POS) tagger followed by word sense retrieval and disambiguation using the WordNet database and the Cyc ontology. After having a preliminary assembly plan, it is executed in simulation to perform a high level validation and also to allow the optimization of the robot movements. If ambiguous or missing information is detected, the system tries to generate a valid assembly plan by analyzing the objects' environment, assembly context and also similar operations stored in its knowledge base.

Named entity recognition can also be useful to identify key information from mission operation orders given to operators of robotics systems, such as Unmaned Aerial Vehicles (UAVs). Highlighting entities such as persons, times, locations, coordinates, targets and organizations allows the human operators to extract the necessary mission information faster. Moreover, in the future it might even be possible to have the robotic system autonomously extract all the required information to carry on the mission without human assistance. The system introduced in [14] was the first step towards this goal, and it was able to extract named entities from textual documents using Condition Random Fields (CRFs) statistical models that relied on features such as word lists, regular expressions, prefixes / sufixes, word case and also unigram / bigram / trigrams models. The evaluation of the NER system was performed using metrics such as precision, recall and accuracy and used 9-fold cross validation for having a rotating train / test dataset in order to avoid model over-fitting.

Several datasets for NER have been presented over the years for news and tweets [15], [16]. This paper aims to provide a dataset for evaluating NER systems with a corpus containing a diverse range of assembly operations for small complex objects (alternators, gearboxes and engines) written in a language discourse that ranges from professional to informal.

III. Dataset sources

This dataset is composed of 10 English instruction manuals with 453 pages detailing assembly operations of alternators, engines and gearboxes (more details shown in Table I). These object categories were selected because they have small, light and diverse components that a typical industrial robot arm can manipulate and also because they have increasing complexity (from the simple gearboxes to the much more complex fuel / steam engines). These manuals were selected for performing NER because they are a representative sample of the several types of manuals that are available for operators working in small parts assembly and also because they were written with a language discourse ranging from very concise and professional to a more colloquial and unstructured type. Moreover they provide tables / lists with the parts and tools required for the assembly operations which are very useful for evaluating NER systems.

Most of these assembly instruction manuals are single column (two of them are dual column) and have Computer Aided Design (CAD) drawings or pictures alongside the assembly procedures. Moreover, some of these procedures are very long, with the description of all the necessary parts for the entire assembly operation while others have the assembly operations split across the main object components.

A. Alternators

Alternators are electrical generators that convert mechanical energy into electrical energy in the form of alternating current. Their assembly is quite complex, involving a lot of small parts and intricate wire bending.

This dataset includes the detailed assembly of two automotive alternators (used to power the electric equipment of cars and charge their battery). One of them was written in a dual

Table I
TOKENIZED DATASET OVERVIEW

	Alternators	*Engines*	*Gearboxes*	*Global*
N^o of pages	84	148	221	453
N^o of assembly procedures	2	40	53	95
N^o of words	9312	22747	31798	63857
N^o of characters	58418	136297	201438	396153

column layout with a lot of diagrams and in a professional and concise language style while the other one was written in single column informal language discourse while using mostly pictures instead of technical diagrams.

B. Gearboxes

Gearboxes are mechanical transmission systems that provide speed and torque conversion while also giving the option of forward and backwards wheel movement. They allow a typical car engine that operates at [600, 7000] Rotations Per Minute (RPM) to move the wheels that usually rotate at [0, 1800] RPM. They can provide more torque when using lower gears and greater speed when employing higher gears. They also give the user more control over the engine performance, allowing better fuel efficiency while also reducing engine wear. Given the high variability in gearbox designs and their interconnecting gears, they can have a complex assembly sequence using mostly medium size parts.

This dataset contains a detailed instruction manual for a car gearbox and another with an extensive collection of small assembly procedures for 52 industrial gearboxes (mainly used in agricultural vehicles such as tractors). Both manuals were written with a professional discourse and in a single column layout. The first had a lot of pictures and CAD drawings, while the second only had technical diagrams for each gearbox assembly procedure.

C. Engines

An engine is a machine designed to convert a given source of energy (such as fuel, electricity, compressed air, elastic / chemical energy, etc) into useful mechanical energy.

In this dataset it is provided an instruction manual with the detailed assembly procedures (35) of a small aircraft engine and also 5 more manuals with the assembly operations of small steam engines. All engine assembly manuals were written with a professional language style and had a single column structure with a lot of accompanying figures.

IV. DATASET PREPARATION

Automatic extraction of text from PDF files with multi-column text, tables and large number of images and diagrams is a challenging task for any NLP system. As such, the dataset preparation included the automatic extraction of text from the PDF files, followed by a manual cleaning and inspection phase in which all the text that was not related to assembly operations was removed. To speedup this process and ensure proper text cleaning across the entire dataset, it was applied a set of regular expressions in order to remove page headers and footers and correct formating issues related with the text extraction. After this preprocessing stage, the dataset was proofread to correct spelling errors. Later on the lists / tables with the information about the required assembly parts / tools was moved into validation files in order to allow the evaluation of a given NER system.

Given that some NLP toolkits such as the Stanford Research Institute Language Models Toolkit (SRILM) [17] expect tokenized text when building N-gram models, a set of regular expressions was applied to the dataset in order to separate the words from the punctuation. Later on, the cleaned text extracted from the PDFs was merged into training (75%) and testing (25%) files according to its respective category, namely the alternators, engines, gearboxes and the entire dataset. These merged files have two versions, one with the original cleaned text and another version with the tokenized text.

V. N-GRAMS LANGUAGE MODELS

An n-gram is a contiguous sequence of items (typically letters or words) that are extracted from an information source (normally text, speech, images or Deoxyribonucleic Acid (DNA)). By counting the n-grams in a knowledge corpus they can be used to create probabilistic language models for predicting the next item given a context. This is useful when developing speech recognizers and Optical Character Recognition (OCR) systems because the n-gram language model can help disambiguate items in the recognition process. Moreover they can be used for implementing text generators or suggestion / auto-complete systems and can also be applied to improve the efficiency of compression / search algorithms.

A. Rank-frequency graphs

Rank-frequency graphs are useful to analyze the word / n-gram diversity of a given text corpus. They are usually plotted in logarithmic scale and have the word rank in the X axis and the word frequency in the Y axis. They are also useful to check if a given text corpus follows the Zipf's law [18], which states that the frequency of a given word in a text corpus is inversely proportional to its rank in the frequency table (as shown in Equation (1)).

$$f(r) \propto \frac{1}{r^{\alpha}} \quad (1)$$

Analyzing Figures 1 to 5 it can be seen that the dataset unigrams to pentagrams follow roughly the Zipf's law. Moreover, the alpha value introduced in Equation (1) that best fits the plotted data starts at 0.25 in the first plot section, then increases to 0.5 in the middle plot section and becomes 1.0 in the last plot section. This is a typical behavior found in most languages [19].

B. Most common and uncommon n-grams

Analyzing the most common and uncommon words / n-grams is useful to gather a quick overview of the topics discussed in a given text corpus.

Table II
TOTAL COUNT OF UNIQUE N-GRAMS IN THE TOKENIZED TRAINING DATASET

N-gram	*Count*
Unigram	2487
Bigram	14140
Trigram	8997
Tetragram	9306
Pentagram	9138

In Table II is shown the unique n-gram counts for the tokenized training dataset. It can be seen that the training dataset vocabulary is composed of 2487 unique words that were used to form 14140 unique bigrams, 8997 unique trigrams, 9306 unique tetragrams and 9138 unique pentagrams.

Looking at Tables III and IV it can be seen that the most common unigrams present in the tokenized training dataset are mainly word articles, prepositions, conjunctions, punctuation and also the beginning and end of sentence tags (<s> and </s>) while the less common unigrams are mainly verbs, nouns and adjectives. Analyzing the Tables V to XII we can see that the word variety and complexity increases as we move from bigrams to pentagrams.

C. N-grams models smoothing

Maximum likelihood estimation gives zero probability to word sequences that have not occurred in the training data. As such, in order to perform tasks such as speech recognition or n-gram perplexity calculation, it is necessary to redistribute some of the probability mass in order to ensure that all the n-grams that can be built with the known vocabulary have non-zero probability. This redistribution of probability mass is known as model smoothing.

One of the simplest smoothing techniques is the additive smoothing, which adds α (typically $\alpha = 1$) to the n-gram counts. More advanced techniques [20] include the Good-Turing smoothing, the backoff algorithms and the interpolated methods. Backoff approaches (such as Katz smoothing), fall back to lower order n-grams probabilities when the higher order n-grams have zero counts. Interpolated algorithms (for example the Jelinek-Mercer, Witten-Bell, Absolute discounting and Kneser-Ney) go a step further and perform a weighted mean of the higher and lower n-gram probabilities.

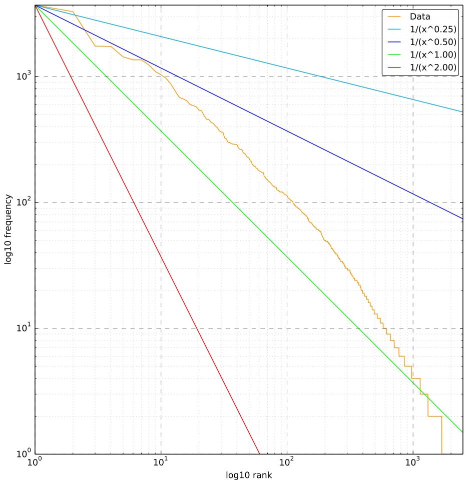

Figure 1. Unigram rank-frequency graph of tokenized dataset

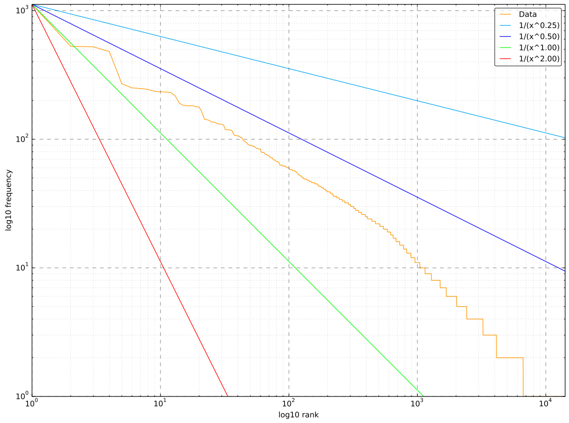

Figure 2. Bigram rank-frequency graph of tokenized dataset

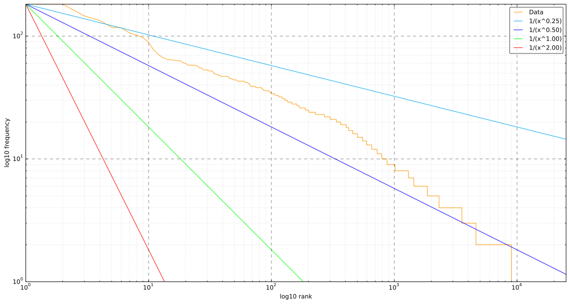

Figure 3. Trigram rank-frequency graph of tokenized dataset

Looking at Table XIII it can be seen that the testing set composed of 20829 words had 4848 out of vocabulary words that caused the occurrence of 1109 zero probabilities in the language models without smoothing (and were ignored when computing the model perplexity).

D. Sentence generation using n-gram models

N-gram models can be used to perform sentence generation by picking a starting n-gram and successively appending the most probable word by taking into account the last $n-1$ added words.

In Sections V-D1 to V-D5 are shown sentences generated using Kneser-Ney interpolated n-gram models built using the tokenized training dataset. Analyzing Section V-D1 it can be seen that a unigram model is not suitable for sentence generation because it does not retain word relations (the sentences are not syntactically correct and there is no coherence of ideas). Bigram models provide some improvements over unigrams, such as local coherence (as can be seen in Section V-D2). However bigram models are not able to generate long mean-

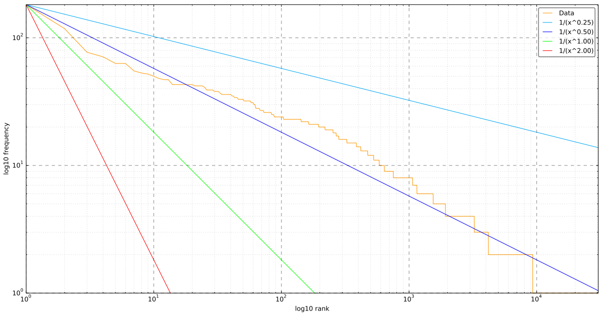

Figure 4. Tetragram rank-frequency graph of tokenized dataset

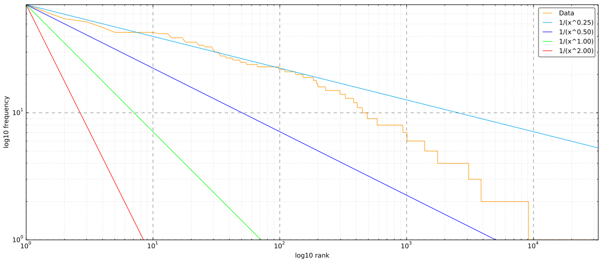

Figure 5. Pentagram rank-frequency graph of tokenized dataset

ingful sentences. Moving to higher order n-grams improves the overall sentence complexity and coherence, as can be seen by analyzing Sections V-D3 to V-D5. However creating higher order n-grams models requires much more processing time and they may still not be able to build syntactically correct and meaningful sentences. As such, n-gram models should be integrated with lexicalized probabilistic context free grammars in order to ensure syntactically correct sentences while retaining the type of language discourse present in the training dataset.

1) Sentences generated using an unigram model:

- <s> the pushrod output magnets . install . attach external been until mower & insulation of do step a then housing shaft ring . 224 better bearings slide bottom . not shims that and play the from always the for though . cylinder it front to : the 0 phase) center voltmeter on . the blade rectifier other) set of bolt . each they in (clean cup engine </s>

2) Sentences generated using a bigram model:

- <s> install input shaft to move shim from table acv - lbs . it . order for 54101) onto output shaft) using shims from notches magnatrons race is seated completely seat on each piston and folded in to nick this is bottomed out , (# 00762520) , try to from they bottom . </s>
- <s> slide outer bearing assembly (item 9) </s>
- <s> the 3 . this time to the side . be (item 7) on each end of out overview of main housing (blade shaft guiding outer race of each column there is correct a 19 input shaft , this till it coating magnetic below . 27) ground other to . order will damage . </s>

Table III
MOST COMMON UNIGRAMS

Unigram	*Count*
the	3692
.	3276
)	1745
(	1735
,	1433
</s>	1357
<s>	1357
of	1235
to	1098
and	1032

Table IV
LEAST COMMON UNIGRAMS

Unigram	*Count*
connected	1
disassembled	1
extend	1
fixed	1
heavy	1
motors	1
path	1
ridges	1
tolerance	1
upwards	1

Table V
MOST COMMON BIGRAMS

Bigram	*Count*
. </s>	1120
of the	528
(#	523
(item	481
<s> install	270
main housing	251
input shaft	248
) .	244
in the	236
from the	234

Table VI
LEAST COMMON BIGRAMS

Bigram	*Count*
engine assembly	1
leads to	1
minor adjustments	1
next step	1
open the	1
put lower	1
rotate it	1
sliding out	1
top gear	1
with care	1

- <s> gearbox main housing against gear and inner bearing cup (. attach the side of grease before deciding gearbox . install without disturbing it is hand , install bolt until discarded and to the long enough for free so the seat in place a . remove shims . do not , install lower bearings have no end play and drift should be required will install on parts . insert the 3 gear back lash should always recheck for proper alignment of the rectifier when this photograph built tapping the cylinder with full wait long enough output shaft in fig . depending upon the steam inlet goes towards you have changed by moving shims (using a snap rings (item 13) onto has end of the 1 . slide bearing area them . place one field then spirits . if wanted includes skip to lap circuit engine . </s>
- <s> check bearing yoke guide support under each lower bearing cup (see line up through the port face towards the bottom . cups in upper bearing cup & cone . </s>
- <s> 1 / 16 ” long end of rolling torque then lock nut with bearing cup & cone . </s>
- <s> note : a 0 snaps - 80 holes manual , if gear side 00769115 dry sand occur while original layshaft nearest the photo) . note : in upper fasteners 00755613) to out put upper hole in the photo) in place with a remove the bearing cone onto input gear spacer using gasket sealer

Table VII
MOST COMMON TRIGRAMS

Trigram	*Count*
& cone)	182
cup & cone	182
) . </s>	146
pre - load	134
bearing pre -	118
bearing cone (	117
bearing cup (	106
end of the	102
into main housing	97
main housing (	89

Table VIII
LEAST COMMON TRIGRAMS

Trigram	*Count*
attach the piston	1
cutting the wires	1
electrical contact is	1
fold the conductor	1
install output shafts	1
proper function of	1
to the airframe	1
using the piston	1
with three screws	1
you push the	1

Table IX
MOST COMMON TETRAGRAMS

Tetragram	*Count*
cup & cone)	182
bearing pre - load	118
bearing cone (#	77
& cone) into	71
bearing cup (#	63
light coat of grease	63
with light coat of	55
pounds of rolling torque	53
check bearing pre -	52
main housing from the	50

Table X
LEAST COMMON TETRAGRAMS

Tetragram	*Count*
center the shaft on	1
connect the wire to	1
cutting the wires to	1
during the attachment of	1
lubricate the cam shaft	1
mount from the side	1
rod in the slot	1
together to make the	1
using the piston rod	1
you slide it into	1

Table XI
MOST COMMON PENTAGRAMS

Pentagram	*Count*
cup & cone) into	71
with light coat of grease	55
check bearing pre - load	52
14 " to 16 "	47
" pounds of rolling torque	43
" to 16 " pounds	43
/ 2 to 1 hour	43
1 / 2 to 1	43
16 " pounds of rolling	43
to 16 " pounds of	43

Table XII
LEAST COMMON PENTAGRAMS

Pentagram	*Count*
area on shaft where seal	1
connect the wire to terminal	1
during the assembly process .	1
facing away from the exhaust	1
it is installed with two	1
lined up with each other	1
put shaft and install the	1
two steam inlets facing each	1
where it fits into the	1
with a screw driver ,	1

Table XIII
TOKENIZED TESTING DATASET OVERVIEW

Metric	*Value*
Sentences	439
Words	20829
Out of vocabulary words ignored	4848
Zero probabilities words ignored	1109

and and and check bearing cup & cone) on the bottom spark plugs from the groove . </s>

- <s> tighten bearing spacer into main housing (do rough or 4 . always be access for oil runs out put upper bearing cone) down between 0 - lbs of the oil a straight regulators spacer to keep bearing is exposed bellcranks to highest is not protrusion with recess outer bearing thrust must end of the are inserted shims) on the side of each cylinder assembly unit 80 nut to your measured opening in photo below 17 . do not , and recheck gear (blade shaft , shaft . </s>
- <s> assembly instructions : 15 in place . the crank timing is against bearing (use the firewall . upper bearing cup try to be 14 to install parts , a small tabs or outward . </s>
- <s> place a gasket shims required will be integral openings positioned of the center of shims (item 17 / 2) on input shaft gear (item 21 engages to remove end hole in the inside bolt together (# 1 hour . sequence shown in the piston to the nomex rear of the engine input shaft , remove all the outside of the female slide the right side of three rectifier bridge (bearing cone (item 24) , and) into inside of the cylinder mount (# 00758668) . </s>

3) Sentences generated using a trigram model:

- <s> install output gear to other to receive the five - in - place insert shaft from gear end of the engine mount to the base together bolt ( ) . 016 " to use same as now nut . you . install shims (# 00758657) input shaft engine it must be moved to rebuild cones on it up through main housing . lower bearing cup (# 00758657) input shaft for end play but back lash between blade shaft (or) , drive cups in till external clip (# 00755613) from rectifier end bearing from </s>
- <s> 6 . take a few 00752140a up magnetic one color have to 10) onto input shaft , slide gear spacer (# 00758656 (11) onto the time to the spindle . to its top port from front bearing cover shims to keep bearing pre - load , if there is no or the readings should indicate continuity , bottom . put a drop of oil in the " a , 24 pinion gear (should have . 017 " to . 019 " , to location for the bearings in the photo) . </s>
- <s> slide the rod and slide it to the laminates to lubricate the end of the long between bolted connections slot . re - install , tighten an disassembly to the screw into the upper bearing cone (# 00758653) into rear of main housing (item 6) . use an awl to of each other . tighten the two columns binding while the using a seal driver of the valve drive block with components if mixture , head are (a) on each piston rod down . </s>
- <s> place the bearing by hand , 11 will damage the case near install) using open tube through the bearing cones can be filled with oil , before deciding gearbox is full wait long enough for oil to run down between output shaft , bearings & gear in position tighten . </s>
- <s> important : the five - in - one screwdriver to where

gear goes . install seal (# 00755615 cup & cone) into bearing , then refill with oil , used trigger wiring </s>

- <s> insert a bushing locktite 272 or wet secure , slot 00757825 while valve stem , gently with side containing the seal in the groove on shaft next to make sure bearing is down . </s>
- <s> check the assembly and insert it into position over key and be using compression hire gears " shims if more is needed add shims till by the lifters in main housing (1 . 017 " to . 019 " back lash . </s>
- <s> looking down into the magnet shoe in for each side before installing input shaft next to rear bearing area . drop input gear (adjustment </s>
- <s> 12 . remove any burrs from the valve spindle this , it should be towards the final drive (# 00758657) onto top of the completed 33 will holding input gear (# 19 of housing (item 24) (fig 4 - lash between blade and tighten down bearing carrier cap (item 8) , coat . snug . </s>
- <s> engage the small hole in bearing caps from one side of the main) from the front . </s>

4) Sentences generated using a tetragram model:

- <s> install input shaft , slide input gear (# 00758645 (ensure test equipment clockwise into the alternator . slide an eccentric strap . guide the male slide bearing and inner bearing cone . </s>
- <s> when this has been done fill gearbox with oil . gearbox can be filled with oil before top cover is installed if wanted . do not install any seals at this point . </s>
- <s> install the lower bearing cone (# 00758650 cup & cone) into rear of main housing . before installing seals . </s>
- <s> install outer bearing cup (# 00758650 cup & cone) on front of the alternator output . all traces of abrasive compound to 25 foot for each other . secure the cylinder to the main bearing until it can threaded increases above oil pump , . assemblies bottomed out put socket wrench makes an the journals retaining nuts & compound over the right few hours the process . is required) from top bearing condition essential dvd screws . install the piston pin pan head screws . </s>
- <s> install top cover (using sealer for gasket , oil plugs and check for leaks , after running mower 1 / 2 to 1 hour check oil level and recheck for leaks . </s>
- <s> slide bearing cone (# 00758650 cup & cone) and inner bearing cup (item 12) , this drives into lower main housing , install shims (item 12) , qty - of the aid of the screws or by subtracting it ’ s into the short arm package included with your trigger shaft housing , . </s>
- <s> install inner bearing (item 3) on the eccentric strap assembly to the input gear (should have the exhaust outlets on use upper reaches and the and check light bolt . near the bearing cup . </s>
- <s> note : a puller can accomplish this task with ease .) . </s>
- <s> set the assembly on top of inner bearing cone (item 22) on shaft next to bearing . </s>
- <s> after assembly , insert 40050 brg strap back 00758663 turning and install steam manifold . </s>

5) Sentences generated using a pentagram model:

- <s> install output shaft (blade shaft) , bearings , gear (output) in horizontal hub housing (# 00762520) . </s>
- <s> when this has been done fill gearbox with oil . gearbox can be filled with a your twelve facets of the core . </s>
- <s> install lower bearing cone (# 00755615 cup & cone) and seal (item 17) onto shaft from gear end . slide inner input shaft & master contact amount regulator bearing (53073 time to 12) . using a metal chips gearbox main housing . </s>
- <s> install bearing cone (# 00755628 cup & cone) into main housing from the bottom . </s>
- <s> fill gearbox with oil , before deciding gearbox is full wait long enough for oil to run down between output shaft bearing , then refill with oil , install top cover (using sealer for gasket) , oil plugs and check for leaks , after running mower 1 / 2 to 1 hour check oil level and recheck for leaks , then with stator output and between rotated . </s>
- <s> repeat checks using insulated heat proceeding . </s>
- <s> insert a bellcrank mount bolts . if it should be gentle on it - ( ) passing the prop hub housing (1) . </s>
- <s> now comes the tricky part . i other now before construction) to slide outer bearing (item 17) in the guides . prior end of the cylinder . </s>
- <s> place the steam chest . </s>
- <s> remove the connecting rod cap . </s>

E. N-gram models evaluation

Language models can be evaluated using an extrinsic or intrinsic approach (chapter 11 of [21]). Extrinsic evaluation provides insight into how well a language model is performing a given task (such as spell correction or speech recognition). Usual metrics for this type of evaluation include precision, accuracy, recall and F1. However, this type of evaluation requires an annotated dataset, and as such, it may be useful to perform a preliminary intrinsic evaluation of the model, in order to assess how well the language model can predict a given test set. The most common metric used to perform intrinsic evaluation is perplexity, which gives the inverse probability of the test set (normalized by the number of words). As such, minimizing perplexity will maximize the probability of correct word prediction when using the given language model.

By analyzing Table XIV we can see that higher order n-grams are better at predicting an unseen test set (they have lower perplexity) and the models that used interpolation as a smoothing technique (Kneser-Ney and Witten-Bell) performed much better than the Laplace add-1 and no smoothing models.

Table XIV
N-GRAMS MODELS PERPLEXITIES

N-gram model	*Perplexity*
Unigram (no smoothing)	243.771
Unigram (Laplace add-1)	245.126
Unigram (Kneser-Ney)	243.771
Unigram (Witten-Bell)	245.126
Bigram (no smoothing)	164.299
Bigram (Laplace add-1)	182.235
Bigram (Kneser-Ney)	56.408
Bigram (Witten-Bell)	56.717
Trigram (no smoothing)	133.965
Trigram (Laplace add-1)	233.030
Trigram (Kneser-Ney)	44.559
Trigram (Witten-Bell)	45.160
Tetragram (no smoothing)	131.832
Tetragram (Laplace add-1)	245.794
Tetragram (Kneser-Ney)	44.067
Tetragram (Witten-Bell)	44.042
Pentagram (no smoothing)	132.059
Pentagram (Laplace add-1)	250.158
Pentagram (Kneser-Ney)	45.725
Pentagram (Witten-Bell)	44.281

VI. CONCLUSIONS

This paper presented the preparation and statistical analysis of a dataset for evaluating NER systems targeted for industrial robotics applications. The dataset contains assembly operations of alternators, gearboxes and engines in textual form and is complemented with object pictures and assembly diagrams. For evaluating NER systems using this dataset, each assembly operation has an associated list with the assembly objects and the quantities needed to successfully perform the product assembly. In order to have a representative dataset, it is provided assembly operations written in a professional and structured manner and also in an informal and colloquial language register. Moreover, it is given a brief statistical analysis of the dataset, with rank-frequency graphs, n-gram models perplexity and also some sentences generated using several n-gram models (from unigrams to pentagrams).

This dataset was built for evaluating NER systems, but can also be used to evaluate information extraction and computer vision systems, given the large textual and image information that it provides for each assembly operation.

VII. FUTURE WORK

Future work for this dataset would include the tagging of all named entities in each assembly operation (instead of having a list for the entire procedure). Moreover, the assembly graph containing both the assembly order and the spatial disposition of the product components would be useful for validating more complex information extraction systems which intend to recover the full assembly knowledge from the text and image representation alone (without operator assistance).

For a system which aims to extract only the named entities in the assembly operations for speeding up object recognition (by restricting the database of object models to search for and recognize), it would be useful to start with the raw text from the PDFs and perform an initial text preprocessing. This stage could include word tokenization, sentence splitting, POS tagging and morphological analysis. Latter on, it could be used a gazetter in conjunction with machine learning algorithms (such as Hidden Markov Models (HMMs) or CRFs) to detect the named entities in the textual assembly operations. After having the named entities, it could be used an orthographic matcher to perform named entity coreference to find different mentions of the same entity and also type disambiguation in order to use word context to make sure that the semantic analysis was correct. The evaluation of a system with these goals can be done by comparing the list of named entities identified as assembly objects with the dataset validation list of product object components. Moreover, if the dataset has named entities tags for each word in the testing dataset, then a more complete evaluation can be done, allowing to assess the reliability of the entity disambiguation and coreference algorithms. Either way, this evaluation would result in the computation of the precision, recall, accuracy and F1 scores for the recognized entities given the list of entities that the NER system was supposed to detect using a k-fold cross validation approach to split the dataset into training and test text.

REFERENCES

[1] B. Akan, A. Ameri, B. Curuklu, and L. Asplund, "Intuitive industrial robot programming through incremental multimodal language and augmented reality," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 3934–3939.

[2] K. Watanabe, C. Jayawardena, and K. Izumi, "Approximate decision making by natural language commands for robots," in *32nd IEEE Annual Conference on Industrial Electronics*, November 2006, pp. 4480–4485.

[3] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox, *Experimental Robotics: The 13th International Symposium on Experimental Robotics*. Springer International Publishing, 2013, ch. Learning to Parse Natural Language Commands to a Robot Control System, pp. 403–415.

[4] E. Neo, T. Sakaguchi, and K. Yokoi, "A humanoid robot that listens, speaks, sees and manipulates in human environments," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, August 2008, pp. 419–425.

[5] P. Barabas, L. Kovacs, and M. Vircikova, "Robot controlling in natural language," in *IEEE 3rd International Conference on Cognitive Infocommunications (CogInfoCom)*, December 2012, pp. 181–186.

[6] L. Derczynski, D. Maynard, G. Rizzo, M. van Erp, G. Gorrell, R. Troncy, J. Petrak, and K. Bontcheva, "Analysis of named entity recognition and linking for tweets," *CoRR*, vol. abs/1410.7182, 2014.

[7] S. Dlugolinsky, M. Ciglan, and M. Laclavik, "Evaluation of named entity recognition tools on microposts," in *IEEE 17th International Conference on Intelligent Engineering Systems (INES)*, June 2013, pp. 197–202.

[8] A. Ekbal, S. Saha, and D. Singh, "Active machine learning technique for named entity recognition," in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*. ACM, 2012, pp. 180–186.

[9] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, ser. CoNLL '09. Association for Computational Linguistics, 2009, pp. 147–155.

[10] R. Al-Rfou, V. Kulkarni, B. Perozzi, and S. Skiena, "POLYGLOT-NER: massive multilingual named entity recognition," *CoRR*, vol. abs/1410.3791, 2014.

[11] M. Rickert, M. E. Foster, M. Giuliani, T. By, G. Panin, and A. Knoll, "Integrating language, vision and action for human robot dialog systems," in *Proceedings of the International Conference on Universal Access in Human-Computer Interaction, HCI International*, ser. Lecture Notes in Computer Science, vol. 4555. Beijing, China: Springer, 2007, pp. 987–995.

[12] M. Stenmark and P. Nugues, "Natural language programming of industrial robots," in *44th International Symposium on Robotics (ISR)*, October 2013, pp. 1–5.

[13] M. Tenorth, D. Nyga, and M. Beetz, "Understanding and executing instructions for everyday manipulation tasks from the world wide web," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2010, pp. 1486–1491.

[14] D. Chesworth, N. Harmon, L. Tanner, S. Guerlain, and M. Balazs, "Named-entity recognition and data visualization techniques to communicate mission command to autonomous systems," in *2016 IEEE Systems and Information Engineering Design Symposium (SIEDS)*, April 2016, pp. 233–238.

[15] M. Dojchinovski and T. Kliegr, "Datasets and gate evaluation framework for benchmarking wikipedia-based ner systems," in *The 12th International Semantic Web Conference (ISWC2013)*, 2013.

[16] M. Röder, R. Usbeck, S. Hellmann, D. Gerber, and A. Both, "N3 - a collection of datasets for named entity recognition and disambiguation in the nlp interchange format," in *The 9th edition of the Language Resources and Evaluation Conference*, 2014.

[17] A. Stolcke, "Srilm - an extensible language modeling toolkit," in *Proceedings International Conference on Spoken Language Processing*, November 2002, pp. 257–286.

[18] S. Piantadosi, "Zipf's word frequency law in natural language: A critical review and future directions," *Psychonomic Bulletin & Review*, vol. 21, no. 5, pp. 1112–1130, 2014.

[19] G. Németh and C. Zainkó, "Multilingual statistical text analysis, zipf's law and hungarian speech generation," *Acta Linguistica Hungarica*, vol. 49, no. 3-4, pp. 385–405, Mar. 2003.

[20] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," Harvard University, Tech. Rep., 1998.

[21] A. Clark, C. Fox, and S. Lappin, *The Handbook of Computational Linguistics and Natural Language Processing*. Wiley-Blackwell, 2010.