Using Integer Linear Programming for Content Selection, Lexicalization, and Aggregation to Produce Compact Texts from OWL Ontologies

Gerasimos Lampouras and **Ion Androutsopoulos**
Department of Informatics
Athens University of Economics and Business
Patission 76, GR-104 34 Athens, Greece
http://nlp.cs.aueb.gr/

Abstract

We present an Integer Linear Programming model of content selection, lexicalization, and aggregation that we developed for a system that generates texts from OWL ontologies. Unlike pipeline architectures, our model jointly considers the available choices in these three text generation stages, to avoid greedy decisions and produce more compact texts. Experiments with two ontologies confirm that it leads to more compact texts, compared to a pipeline with the same components, with no deterioration in the perceived quality of the generated texts. We also present an approximation of our model, which allows longer texts to be generated efficiently.

1 Introduction

Concept-to-text natural language generation (NLG) generates texts from formal knowledge representations (Reiter and Dale, 2000). With the emergence of the Semantic Web (Berners-Lee et al., 2001; Shadbolt et al., 2006; Antoniou and van Harmelen, 2008), interest in concept-to-text NLG has been revived and several methods have been proposed to express axioms of OWL ontologies (Grau et al., 2008), a form of description logic (Baader et al., 2002), in natural language (Bontcheva, 2005; Mellish and Sun, 2006; Galanis and Androutsopoulos, 2007; Mellish and Pan, 2008; Schwitter et al., 2008; Schwitter, 2010; Liang et al., 2011; Williams et al., 2011).

NLG systems typically employ a pipeline architecture. They usually start by selecting the logical facts (axioms, in the case of an OWL ontology) to be expressed. The purpose of the next stage, text planning, ranges from simply ordering the facts to be expressed to making more complex decisions about the rhetorical structure of the text. Lexicalization then selects the words and syntactic structures that will realize each fact, specifying how each fact can be expressed as a single sentence. Sentence aggregation may then combine shorter sentences to form longer ones. Another component generates appropriate referring expressions, and surface realization produces the final text.

Each stage of the pipeline is treated as a local optimization problem, where the decisions of the previous stages cannot be modified. This arrangement produces texts that may not be optimal, since the decisions of the stages have been shown to be co-dependent (Danlos, 1984; Marciniak and Strube, 2005; Belz, 2008). For example, decisions made during content selection may maximize importance measures, but may produce facts that are difficult to turn into a coherent text; also, content selection and lexicalization may lead to more or fewer sentence aggregation opportunities. Some of these problems can be addressed by overgenerating at each stage (e.g., producing several alternative sets of facts at the end of content selection, several alternative lexicalizations etc.) and employing a final ranking component to select the best combination (Walker et al., 2001). This overgenerate and rank approach, however, may also fail to find an optimal solution, and it generates an exponentially large number of candidate solutions when several components are pipelined.

In this paper, we present an Integer Linear Programming (ILP) model that combines content selection, lexicalization, and sentence aggregation. Our model does not consider directly text planning, nor referring expression generation, which we hope to include in future work, but it is combined with an external simple text planner and an external referring expression generation component; we also do not discuss surface realization. Unlike pipeline architectures, our model jointly examines the possible choices in the three NLG stages it considers, to avoid greedy local decisions.

Proceedings of the 14th European Workshop on Natural Language Generation, pages 51–60,
Sofia, Bulgaria, August 8-9 2013.

Given an individual (entity) or class of an OWL ontology and a set of facts (axioms) about the individual or class, we aim to produce a compact text that expresses as many facts in as few words as possible. This is desirable when space is limited or expensive, e.g., when displaying product descriptions on smartphones, or when including advertisements in Web search results. If an importance score is available for each fact, our model can take it into account to prefer expressing important facts, again using as few words as possible. The model itself, however, does not produce importance scores, i.e., we assume that the scores are produced by a separate process (Barzilay and Lapata, 2005; Demir et al., 2010), not included in our content selection. In the experiments of this article, we treat all the facts as equally important.

Although the search space of our model is very large and ILP problems are in general NP-hard, off-the-shelf ILP solvers can be used, which can be very fast in practice and guarantee finding a global optimum. Experiments with two ontologies show that our ILP model outperforms, in terms of expressed facts per word, an NLG system that uses the same components connected in a pipeline, with no deterioration in perceived text quality; the ILP model may actually lead to texts of higher quality, compared to those of the pipeline, when there are many facts to express. We also present an approximation of our ILP model, which is more efficient when larger numbers of facts need to be expressed.

Section 2 discusses previous related work. Section 3 defines our ILP model. Section 4 presents our experimentals. Section 5 concludes.

2 Related work

Marciniak and Strube (2005) propose a general ILP approach for language processing applications where the decisions of classifiers that consider particular, but co-dependent, subtasks need to be combined. They also show how their approach can be used to generate multi-sentence route directions, in a setting with very different inputs and processing stages than the ones we consider.

Barzilay and Lapata (2005) treat content selection as an optimization problem. Given a pool of facts and scores indicating their importance, they select the facts to express by formulating an optimization problem similar to energy minimization. The problem is solved by applying a minimal cut partition algorithm to a graph representing the pool of facts and the importance scores. The importance scores of the facts are obtained via supervised machine learning (AdaBoost) from a dataset of (sports) facts and news articles expressing them.

In other work, Barzilay and Lapata (2006) consider sentence aggregation. Given a set of facts that a content selection stage has produced, aggregation is viewed as the problem of partitioning the facts into optimal subsets. Sentences expressing facts of the same subset are aggregated to form a longer sentence. The optimal partitioning maximizes the pairwise similarity of the facts in each subset, subject to constraints that limit the number of subsets and the number of facts in each subset. A Maximum Entropy classifier predicts the semantic similarity of each pair of facts, and an ILP model is used to find the optimal partitioning.

Althaus et al. (2004) show that ordering a set of sentences to maximize local coherence is equivalent to the traveling salesman problem and, hence, NP-complete. They also show an ILP formulation of the problem, which can be solved efficiently in practice using branch-and-cut with cutting planes.

Kuznetsova et al. (2012) use ILP to generate image captions. They train classifiers to detect the objects in each image. Having identified the objects of a given image, they retrieve phrases from the captions of a corpus of images, focusing on the captions of objects that are similar (color, texture, shape) to the ones in the given image. To select which objects of the image to report and in what order, Kuznetsova et al. maximize (via ILP) the mean of the confidence scores of the object detection classifiers and the sum of the co-occurrence probabilities of the objects that will be reported in adjacent positions in the caption. Having decided which objects to report and their order, Kuznetsova et al. use a second ILP model to decide which phrases to use for each object and to order the phrases. The second ILP model maximizes the confidence of the phrase retrieval algorithm and the local cohesion between subsequent phrases.

Joint optimization ILP models have also been used in multi-document text summarization and sentence compression (McDonald, 2007; Clarke and Lapata, 2008; Berg-Kirkpatrick et al., 2011; Galanis et al., 2012; Woodsend and Lapata, 2012), where the input is text, not formal knowledge represetations. Statistical methods to jointly perform content selection, lexicalization, and surface realization have also been proposed in NLG (Liang et

al., 2009; Konstas and Lapata, 2012a; Konstas and Lapata, 2012b), but they are currently limited to generating single sentences from flat records, as opposed to ontologies. Our method is the first one to consider content selection, lexicalization, and sentence aggregation as an ILP joint optimization problem in the context of multi-sentence concept-to-text generation.

3 Our ILP model of NLG

Let $F = \{f_1, \ldots, f_n\}$ be the set of all the facts f_i (OWL axioms) about the individual or class to be described. OWL axioms can be represented as sets of RDF triples of the form $\langle S, R, O \rangle$, where S is an individual or class, O is another individual, class, or datatype value, and R is a relation (property) that connects S to O.[1] Hence, we can assume that each fact f_i is a triple $\langle S_i, R_i, O_i \rangle$.[2]

For each fact f_i, a set $P_i = \{p_{i1}, p_{i2}, \ldots\}$ of alternative sentence plans is available. Each sentence plan p_{ik} specifies how to express $f_i = \langle S_i, R_i, O_i \rangle$ as an alternative single sentence. In our work, a sentence plan is a sequence of slots, along with instructions specifying how to fill the slots in; and each sentence plan is associated with the relations it can express. For example, $\langle$`exhibit12`, `foundIn`, `athens`$\rangle$ could be expressed using a sentence plan like "[*ref*(S)] [find$_{past}$] [in] [*ref*(O)]", where square brackets denote slots, *ref*(S) and *ref*(O) are instructions requiring referring expressions for S and O in the corresponding slots, and "find$_{past}$" requires the simple past form of "find". In our example, the sentence plan would lead to a sentence like "Exhibit 12 was found in Athens". We call *elements* the slots with their instructions, but with "S" and "O" accompanied by the individuals, classes, or datatype values they refer to; in our example, the elements are "[*ref*(S: `exhibit12`)]", "[find$_{past}$]", "[in]", "[*ref*(O: `athens`)]".

Different sentence plans may lead to more or fewer aggregation opportunities; e.g., sentences with the same verb are easier to aggregate. We use aggregation rules similar to those of Dalianis (1999), which operate on sentence plans and usually lead to shorter texts, as in the example below.

> Bancroft Chardonnay is a kind of Chardonnay. It is made in Bancroft. ⇒ Bancroft Chardonnay is a kind of Chardonnay made in Bancroft.

Let $s_1, \ldots, s_m$ be disjoint subsets of F, each containing 0 to n facts, with $m < n$. A single sentence is generated for each subset s_j by aggregating the sentences (more precisely, the sentence plans) expressing the facts of s_j.[3] An empty s_j generates no sentence, i.e., the resulting text can be at most m sentences long. Let us also define:

$$a_i = \begin{cases} 1, & \text{if fact } f_i \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$l_{ikj} = \begin{cases} 1, & \text{if sentence plan } p_{ik} \text{ is used to express} \\ & \text{fact } f_i, \text{ and } f_i \text{ is in subset } s_j \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$b_{tj} = \begin{cases} 1, & \text{if element } e_t \text{ is used in subset } s_j \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

and let B be the set of all the distinct elements (no duplicates) from all the available sentence plans that can express the facts of F. The length of an aggregated sentence resulting from a subset s_j can be roughly estimated by counting the distinct elements of the sentence plans that have been chosen to express the facts of s_j; elements that occur more than once in the chosen sentence plans of s_j are counted only once, because they will probably be expressed only once, due to aggregation.

Our objective function (4) maximizes the total importance of the selected facts (or simply the number of selected facts, if all facts are equally important), and minimizes the number of distinct elements in each subset s_j, i.e., the approximate length of the corresponding aggregated sentence; an alternative explanation is that by minimizing the number of distinct elements in each s_j, we favor subsets that aggregate well. By a and b we jointly denote all the a_i and b_{tj} variables. The two parts of the objective function are normalized to $[0, 1]$ by dividing by the total number of available facts $|F|$ and the number of subsets m times the total number of distinct elements $|B|$. We assume that the importance scores $imp(f_i)$ are provided by a separate component (Barzilay and Lapata, 2005; Demir et al., 2010) and range in $[0, 1]$. The parameters λ_1, λ_2 are used to tune the priority given to expressing many important facts vs.

[1] See `www.w3.org/TR/owl2-mapping-to-rdf/`.

[2] We actually convert the RDF triples to simpler *message triples*, so that each message triple can be easily expressed by a simple sentence, but we do not discuss this conversion here.

[3] All the sentences of every possible subset s_j can be aggregated, because all the sentences share the same subject, the class or individual being described. If multiple aggregation rules apply, we use the one that leads to a shorter text.

generating shorter texts; we set $\lambda_1 + \lambda_2 = 1$.

$$\max_{a,b} \lambda_1 \cdot \sum_{i=1}^{|F|} \frac{a_i \cdot imp(f_i)}{|F|} - \lambda_2 \cdot \sum_{j=1}^{m} \sum_{t=1}^{|B|} \frac{b_{tj}}{m \cdot |B|} \quad (4)$$

subject to:

$$a_i = \sum_{j=1}^{m} \sum_{k=1}^{|P_i|} l_{ikj}, \text{for } i = 1, \ldots, n \quad (5)$$

$$\sum_{e_t \in B_{ik}} b_{tj} \geq |B_{ik}| \cdot l_{ikj}, \text{for } \begin{matrix} i = 1, \ldots, n \\ j = 1, \ldots, m \\ k = 1, \ldots, |P_i| \end{matrix} \quad (6)$$

$$\sum_{p_{ik} \in P(e_t)} l_{ikj} \geq b_{tj}, \text{for } \begin{matrix} t = 1, \ldots, |B| \\ j = 1, \ldots, m \end{matrix} \quad (7)$$

$$\sum_{t=1}^{|B|} b_{tj} \leq B_{max}, \text{for } j = 1, \ldots, m \quad (8)$$

$$\sum_{k=1}^{|P_i|} l_{ikj} + \sum_{k'=1}^{|P_{i'}|} l_{i'k'j} \leq 1, \text{for } \begin{matrix} j = 1, \ldots, m,\ i = 2, \ldots, n \\ i' = 1, \ldots, n-1; i \neq i' \\ \textit{section}(f_i) \neq \textit{section}(f_{i'}) \end{matrix} \quad (9)$$

Constraint 5 ensures that for each selected fact, only one sentence plan in only one subset is selected; if a fact is not selected, no sentence plan for the fact is selected either. $|\sigma|$ denotes the cardinality of a set σ. In constraint 6, B_{ik} is the set of distinct elements e_t of the sentence plan p_{ik}. This constraint ensures that if p_{ik} is selected in a subset s_j, then all the elements of p_{ik} are also present in s_j. If p_{ik} is not selected in s_j, then some of its elements may still be present in s_j, if they appear in another selected sentence plan of s_j.

In constraint 7, $P(e_t)$ is the set of sentence plans that contain element e_t. If e_t is used in a subset s_j, then at least one of the sentence plans of $P(e_t)$ must also be selected in s_j. If e_t is not used in s_j, then no sentence plan of $P(e_t)$ may be selected in s_j. Lastly, constraint 8 limits the number of elements that a subset s_j can contain to a maximum allowed number B_{max}, in effect limiting the maximum length of an aggregated sentence.

We assume that each relation R has been manually mapped to a single *topical section*; e.g., relations expressing the color, body, and flavor of a wine may be grouped in one section, and relations about the wine's producer in another. The section of a fact $f_i = \langle S_i, R_i, O_i \rangle$ is the section of its relation R_i. Constraint 9 ensures that facts from different sections will not be placed in the same subset s_j, to avoid unnatural aggregations.

4 Experiments

We used NaturalOWL (Galanis and Androutsopoulos, 2007; Galanis et al., 2009; Androutsopoulos et al., 2013), an NLG system for OWL ontologies that relies on a pipeline of content selection, text planning, lexicalization, aggregation, referring expression generation, and surface realization components.[4] We modified the content selection, lexicalization, and aggregation components to use our ILP model, maintaining the aggregation rules of the original system. For referring expressions and surface realization, the new system, called ILPNLG, invokes the corresponding components of the original system. We use branch-and-cut to solve the ILP problems.[5]

The original system, hereafter called PIPELINE, assumes that each relation has been mapped to a topical section, as in ILPNLG. It also assumes that a manually specified order of the sections and the relations of each section is available, which is used by the text planner to order the selected facts (by their relations). The subsequent components of the pipeline are not allowed to change the order of the facts, and aggregation operates only on sentence plans of adjacent facts from the same section. In ILPNLG, the manually specified order of sections and relations is used to order the sentences of each subset s_j (before aggregating them), the aggregated sentences in each section (each aggregated sentence inherits the minimum order of its constituents), and the sections (with their sentences).

4.1 Experiments with the Wine Ontology

In a first set of experiments, we used the Wine Ontology, which had also been used in previous experiments with PIPELINE (Androutsopoulos et al., 2013). The ontology contains 63 wine classes, 52 wine individuals, a total of 238 classes and individuals (including wineries, regions, etc.), and 14 properties.[6] We kept the 2 topical sections, the ordering of sections and relations, and the sentence plans of the previous experiments, but we added more sentence plans to ensure that 3 sentence plans were available per relation. We generated English texts for the 52 wine individuals

[4] All the software and data that we used will be freely available from `http://nlp.cs.aueb.gr/software.html`. We use version 2 of NaturalOWL.

[5] We use the branch-and-cut implementation of GLPK with mixed integer rounding, mixed cover, and clique cuts; see `sourceforge.net/projects/winglpk/`.

[6] See `www.w3.org/TR/owl-guide/wine.rdf`.

of the ontology; we did not experiment with texts describing classes, because we could not think of multiple alternative sentence plans for many of their axioms. For each wine individual, there were 5 facts on average and a maximum of 6 facts. We set the importance scores $imp(f_i)$ of all the facts f_i to 1, to make the decisions of PIPELINE and ILPNLG easier to understand; both systems use the same importance scores. PIPELINE does not provide any mechanism to estimate the importance scores, assuming that they are provided manually.

PIPELINE has a parameter M specifying the maximum number of facts it is allowed to report per text. When M is smaller than the number of available facts ($|F|$) and all the facts are treated as equally important, as in our experiments, it selects randomly M of the available facts. We repeated the generation of PIPELINE's texts for the 52 individuals for $M = 2, 3, 4, 5, 6$. For each M, the texts of PIPELINE for the 52 individuals were generated three times, each time using one of the different alternative sentence plans of each relation. We also generated the texts using a variant of PIPELINE, dubbed PIPELINESHORT, which always selects the shortest (in elements) sentence plan among the available ones. In all cases, PIPELINE and PIPELINESHORT were allowed to form aggregated sentences containing up to $B_{max} = 22$ distinct elements, which was the number of distinct elements of the longest aggregated sentence in the previous experiments (Androutsopoulos et al., 2013), where PIPELINE was allowed to aggregate up to 3 original sentences.[7]

With ILPNLG, we repeated the generation of the texts of the 52 individuals using different values of λ_1 ($\lambda_2 = 1 - \lambda_1$), which led to texts expressing from zero to all of the available facts. We set the maximum number of fact subsets to $m = 3$, which was the maximum number of (aggregated) sentences in the texts of PIPELINE and PIPELINESHORT. Again, we set $B_{max} = 22$.

We compared ILPNLG to PIPELINE and PIPELINESHORT by measuring the average number of facts they reported divided by the average text length (in words). Figure 1 shows this ratio as a function of the average number of reported facts, along with 95% confidence intervals (of sample means). PIPELINESHORT achieved better results than PIPELINE, but the differences were small.

For $\lambda_1 < 0.2$, ILPNLG produces empty texts,

[7] We modified the two pipeline systems to count elements.

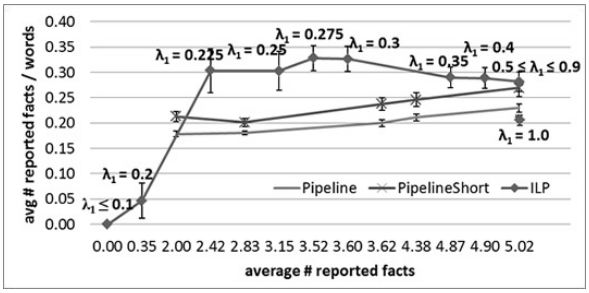

Figure 1: Facts/words of Wine Ontology texts.

because it focuses on minimizing the number of distinct elements of each text. For $\lambda_1 \geq 0.225$, it performs better than the other systems. For $\lambda_1 \approx 0.3$, it obtains the highest fact/words ratio by selecting the facts and sentence plans that lead to the most compressive aggregations. For greater values of λ_1, it selects additional facts whose sentence plans do not aggregate that well, which is why the ratio declines. For small numbers of facts, the two pipeline systems select facts and sentence plans that offer few aggregation opportunities; as the number of selected facts increases, some more aggregation opportunities arise, which is why the facts/words ratio of the two systems improves. In all the experiments, the ILP solver was very fast (average: 0.08 sec, worst: 0.14 sec per text).

We show below texts produced by PIPELINE ($M = 4$) and ILPNLG ($\lambda_1 = 0.3$).

PIPELINE: This is a strong Sauternes. It is made from Semillon grapes and it is produced by Chateau D'ychem.

ILPNLG: This is a strong Sauternes. It is made from Semillon grapes by Chateau D'ychem.

PIPELINE: This is a full Riesling and it has moderate flavor. It is produced by Volrad.

ILPNLG: This is a full sweet moderate Riesling.

In the first pair, PIPELINE uses different verbs for the grapes and producer, whereas ILPNLG uses the same verb, which leads to a more compressive aggregation; both texts describe the same wine and report 4 facts. In the second pair, ILPNLG has chosen to express the sweetness instead of the producer, and uses the same verb ("be") for all the facts, leading to a shorter sentence; again both texts describe the same wine and report 4 facts. In both examples, some facts are not aggregated because they belong in different sections.

We also wanted to investigate the effect that the higher facts/words ratio of ILPNLG has on the perceived quality of the generated texts, compared to the texts of the pipeline. We were concerned that the more compressive aggregations of ILPNLG

Criteria	PIPELINESHORT	ILPNLG
Sentence fluency	4.75 ± 0.21	**4.85** ± 0.10
Text structure	**4.94** ± 0.06	4.88 ± 0.14
Clarity	**4.77** ± 0.18	4.75 ± 0.15
Overall	4.52 ± 0.20	**4.60** ± 0.18

Table 1: Human scores for Wine Ontology texts.

might lead to sentences that sound less fluent or unnatural, though aggregation often helps produce more natural texts. We were also concerned that the more compact texts of ILPNLG might be perceived as being more difficult to understand (less clear) or less well-structured. To investigate these issues, we showed the $52 \times 2 = 104$ texts of PIPELINESHORT ($M = 4$) and ILPNLG ($\lambda_1 = 0.3$) to 6 computer science students not involved in the work of this article; they were all fluent, though not native, English speakers. Each one of the 104 texts was given to exactly one student. Each student was given approximately 9 randomly selected texts of each system. The OWL statements that the texts were generated from were not shown, and the students did not know which system had generated each text. Each student was shown all of his/her texts in random order, regardless of the system that generated them. The students were asked to score each text by stating how strongly they agreed or disagreed with statements S_1–S_3 below. A scale from 1 to 5 was used (1: strong disagreement, 3: ambivalent, 5: strong agreement).

(S_1) *Sentence fluency*: The sentences of the text are fluent, i.e., each sentence *on its own* is grammatical and sounds natural. When two or more smaller sentences are combined to form a single, longer sentence, the resulting longer sentence is also grammatical and sounds natural.

(S_2) *Text structure*: The order of the sentences is appropriate. The text presents information by moving reasonably from one topic to another.

(S_3) *Clarity*: The text is easy to understand, provided that the reader is familiar with basic wine terms.

The students were also asked to provide an overall score (1–5) per text. We did not score referring expressions, since both systems use the same component to generate them.

Table 1 shows the average scores of the two systems with 95% confidence intervals (of sample means). For each criterion, the best score is shown in bold. The sentence fluency and overall scores of ILPNLG are slightly higher than those of PIPELINESHORT, whereas PIPELINESHORT obtained a slightly higher score for text structure and clarity. The differences, however, are very small, especially in clarity, and there is no statistically significant difference between the two systems in any of the criteria.[8] Hence, there was no evidence in these experiments that the highest facts/words ratio of ILPNLG comes at the expense of lower perceived text quality. We investigated these issues further in a second set of experiments, discussed next, where the generated texts were longer.

4.2 Consumer Electronics experiments

In the second set of experiments, we used the Consumer Electronics Ontology, which had also been used in previous work with PIPELINE. The ontology comprises 54 classes and 441 individuals (e.g., printer types, paper sizes), but no information about particular products.[9] In previous work, 30 individuals (10 digital cameras, 10 camcorders, 10 printers) were added to the ontology; they were randomly selected from a publicly available dataset of 286 digital cameras, 613 camcorders, and 58 printers, whose instances comply with the Consumer Electronics Ontology.[10] We kept the 6 topical sections, the ordering of sections and relations, and the sentence plans of the previous work, but we added more sentence plans to ensure that 3 sentence plans were available for almost every relation; for some relations we could not think of enough sentence plans. Again, we set the importance scores of all the facts to 1.

We generated texts with PIPELINE and PIPELINESHORT for the 30 individuals, for $M = 3, 6, 9, \ldots, 21$. Again for each M, the texts of PIPELINE were generated three times, each time using one of the different alternative sentence plans of each relation. PIPELINE and PIPELINESHORT were allowed to form aggregated sentences containing up to $B_{max} = 39$ distinct elements, which was the number of distinct elements of the longest aggregated sentence in the previous work with this ontology, where PIPELINE was allowed to aggregate up to 3 original sentences. We also set $B_{max} = 39$ in ILPNLG.

There are 14 facts (F) on average and a maximum of 21 facts for each one of the 30 individuals, compared to the 5 facts on average and the maximum of 6 facts of the experiments with the Wine Ontology. Hence, the texts of the Consumer

[8] The confidence intervals do not overlap, and we also performed paired two-tailed t-tests ($\alpha = 0.05$) to check for statistical significance. In previous work, where judges were asked to score texts using the same criteria, inter-annotator agreement was strong (sample Pearson correlation $r \geq 0.91$).

[9] Ontology available from `www.ebusiness-unibw.org/ontologies/consumerelectronics/v1`.

[10] See `rdf4ecommerce.esolda.com/`.

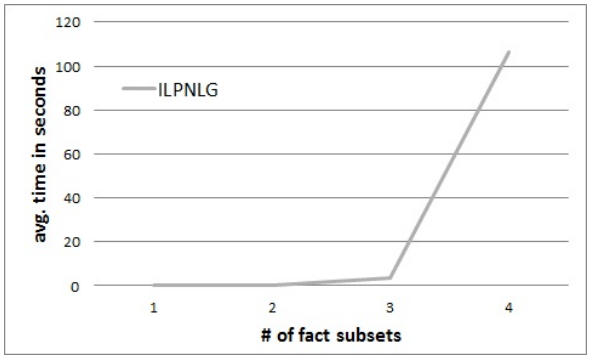

Figure 2: Average solver times for ILPNLG for different maximum numbers of fact subsets (m).

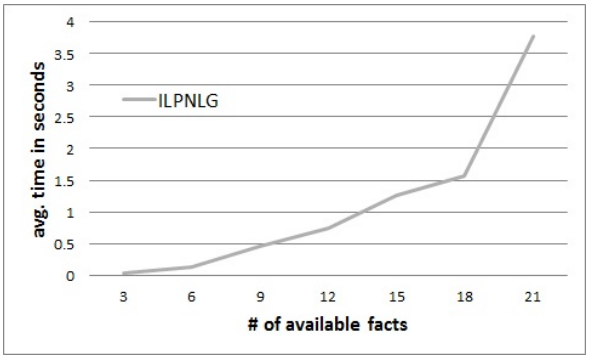

Figure 3: Average solver times for ILPNLG for different numbers of available facts ($|F|$) and $m = 3$.

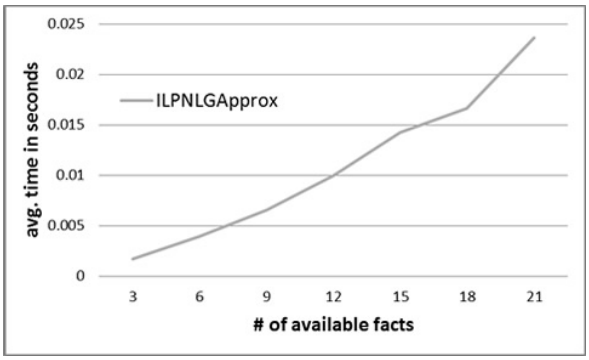

Figure 4: Avg. solver times for ILPNLGAPPROX for different max. numbers of fact subsets (m).

Electronics Ontology are much longer, when they report all the available facts. To generate texts for the 30 individuals with ILPNLG, we would have to set the maximum number of fact subsets to $m = 10$, which was the maximum number of (aggregated) sentences in the texts of PIPELINE and PIPELINESHORT. The number of variables of our ILP model, however, grows exponentially to m and the number of available facts $|F|$. Figure 2 shows the average time the ILP solver took for different values of m in the experiments with the Consumer Electronics ontology; the results are also averaged for $\lambda_1 = 0.4, 0.5, 0.6$ ($\lambda_2 = 1 - \lambda_1$). For $m = 4$, the solver took 1 minute and 47 seconds on average per text; recall that $|F|$ is also much larger now, compared to the experiments of the previous section. For $m = 5$, the solver was so slow that we aborted the experiment. Figure 3 shows the average solver time for different numbers of available facts $|F|$, for $m = 3$; in this case, we modified the set of available facts (F) of every individual to contain $3, 6, 9, 12, 15, 18, 21$ facts; the results are averaged for $\lambda_1 = 0.4, 0.5, 0.6$. Although the times of Fig. 3 also grow exponentially, they remain under 4 seconds, showing that the main problem for ILPNLG is m, the number of fact subsets, which is also the maximum allowed number of (aggregated) sentences of each text.

To be able to efficiently generate texts with larger m values, we use a variant of ILPNLG, called ILPNLGAPPROX, which considers each fact subset separately. ILPNLGAPPROX starts with the full set of available facts (F) and uses our ILP model (Section 3) with $m = 1$ to produce the first (aggregated) sentence of the text. It then removes the facts expressed by the first (aggregated) sentence from F, and uses the ILP model, again with $m = 1$, to produce the second (aggregated) sentence etc. This process is repeated until we produce the maximum number of allowed aggregated sentences, or until we run out of facts. ILPNLGAPPROX is an approximation of ILPNLG, in the sense that it does not consider all the fact subsets jointly and, hence, does not guarantee finding a globally optimal solution for the entire text. Figures 4–5 show the average solver times of ILPNLGAPPROX for different values of m and $|F|$; all the other settings are as in Figures 2–3. The solver times of ILPNLGAPPROX grow approximately linearly to m and $|F|$ and are under 0.3 seconds in all cases.

Figure 6 shows the average facts/words ratio of ILPNLGAPPROX ($m = 10$), PIPELINE and PIPELINESHORT, along with 95% confidence intervals (of sample means), for the texts of the 30 individuals. Again, PIPELINESHORT achieves slightly better results than PIPELINE, but the differences are now smaller (cf. Fig. 1). ILPNLGAPPROX behaves very similarly to ILPNLG in the Wine Ontology experiments (cf. Fig. 1); for $\lambda_1 \leq 0.35$, it produces empty texts, while for $\lambda_1 \geq 0.4$ it performs better than the other systems. ILPNLGAPPROX obtains

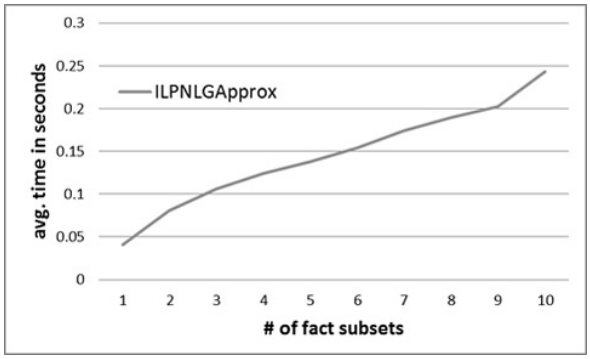

Figure 5: Avg. solver times for ILPNLGAPPROX for different $|F|$ values and $m = 3$.

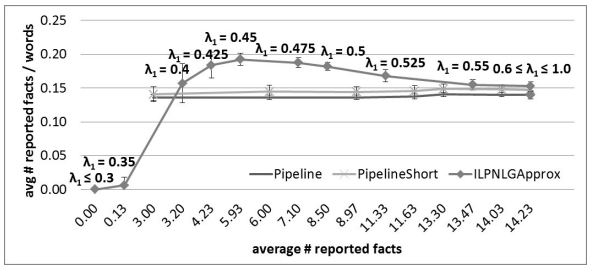

Figure 6: Facts/words for Consumer Electronics.

the highest facts/words ratio for $\lambda_1 = 0.45$, where it selects the facts and sentence plans that lead to the most compressive aggregations. For greater values of λ_1, it selects additional facts whose sentence plans do not aggregate that well, which is why the ratio declines. The two pipeline systems select facts and sentence plans that offer very few aggregation opportunities; as the number of selected facts increases, some more aggregation opportunities arise, which is why the facts/words ratio of the two systems improves slightly, though the improvement is now hardly noticeable.

We show below two example texts produced by PIPELINE ($M = 6$) and ILPNLGAPPROX ($\lambda_1 = 0.45$). Both texts report 6 facts, but ILPNLGAPPROX has selected facts and sentence plans that allow more compressive aggregations. Recall that we treat all the facts as equally important.

PIPELINE: Sony DCR-TRV270 requires minimum illumination of 4.0 lux and its display is 2.5 in. It features a sports scene mode, it includes a microphone and an IR remote control. Its weight is 780.0 grm.

ILPNLGAPPROX: Sony DCR-TRV270 has a microphone and an IR remote control. It is 98.0 mm high, 85.0 mm wide, 151.0 mm deep and it weighs 780.0 grm.

We showed the $30 \times 2 = 60$ texts of PIPELINESHORT ($M = 6$) and ILPNLGAPPROX ($\lambda_1 = 0.45$) to the same 6 students, as in Section 4.1. Again, each text was given to exactly one student. Each student was given approximately 5 randomly selected texts of each system. The OWL statements that the texts were generated from were not shown, and the students did not know which system had generated each text. Each student was shown all of his/her texts in random order, regardless of the system that generated them. The students were asked to score each text by stating how strongly they agreed or disagreed with statements S_1–S_3, as in Section 4.1. They were also asked to provide an overall score (1–5) per text.

Criteria	PIPELINESHORT	ILPNLGAPPROX
Sentence fluency	4.50 ± 0.30	$\mathbf{4.87} \pm 0.12$
Text structure	4.33 ± 0.36	$\mathbf{4.73} \pm 0.22$
Clarity	4.53 ± 0.29	$\mathbf{4.97 \pm 0.06}$
Overall	4.10 ± 0.31	$\mathbf{4.73 \pm 0.16}$

Table 2: Human scores for Consumer Electronics.

Table 2 shows the average scores of the two systems with 95% confidence intervals (of sample means). For each criterion, the best score is shown in bold; the confidence interval of the best score is also shown in bold, if it does not overlap with the confidence interval of the other system. Unlike the Wine Ontology experiments, the scores of our ILP approach are now higher than those of the pipeline in all of the criteria, and the differences are also larger, though the differences are statistically significant only for clarity and overall quality.[11] We attribute these differences to the fact that the texts are now longer and the sentence plans more varied, which often makes the texts of the pipeline sound verbose and, hence, more difficult to follow, compared to the more compact texts of ILPNLGAPPROX, which sound more concise.

Overall, the human scores of the experiments with the two ontologies suggest that the higher facts/words ratio of our ILP approach does *not* come at the expense of lower perceived text quality. On the contrary, the texts of the ILP approach may be perceived as clearer and overall better than those of the pipeline, when the texts are longer.

5 Conclusions

We presented an ILP model of content selection, lexicalization, and aggregation that jointly considers the possible choices in the three stages, to

[11] When two confidence intervals do not overlap, the difference is statistically significant. When they overlap, the difference may still be statistically significant; we performed additional paired two-tailed t-tests ($\alpha = 0.05$) in those cases.

avoid greedy local decisions and produce more compact texts. The model has been embedded in NaturalOWL, a NLG system for OWL ontologies, which used a pipeline architecture in its original form. Experiments with two ontologies confirmed that our approach leads to expressing more facts per word, with no deterioration in the perceived text quality; the ILP approach may actually lead to texts perceived as clearer and overall better, compared to the pipeline, when there are many facts to express. We also presented an approximation of our ILP model, which allows longer texts to be generated efficiently. We plan to extend our model to include text planning, referring expression generation, and mechanisms to obtain importance scores.

Acknowledgments

This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) – Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.

References

E. Althaus, N. Karamanis, and A. Koller. 2004. Computing locally coherent discourses. In *42nd Annual Meeting of ACL*, pages 399–406, Barcelona, Spain.

I. Androutsopoulos, G. Lampouras, and D. Galanis. 2013. Generating natural language descriptions from OWL ontologies: the NaturalOWL system. Technical report, Natural Language Processing Group, Department of Informatics, Athens University of Economics and Business.

G. Antoniou and F. van Harmelen. 2008. *A Semantic Web primer*. MIT Press, 2nd edition.

F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors. 2002. *The Description Logic Handbook*. Cambridge Univ. Press.

R. Barzilay and M. Lapata. 2005. Collective content selection for concept-to-text generation. In *HLT-EMNLP*, pages 331–338, Vancouver, BC, Canada.

R. Barzilay and M. Lapata. 2006. Aggregation via set partitioning for natural language generation. In *HLT-NAACL*, pages 359–366, New York, NY.

A. Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431–455.

T. Berg-Kirkpatrick, D. Gillick, and D. Klein. 2011. Jointly learning to extract and compress. In *49th Meeting of ACL*, pages 481–490, Portland, OR.

T. Berners-Lee, J. Hendler, and O. Lassila. 2001. The Semantic Web. *Scientific American*, May:34–43.

K. Bontcheva. 2005. Generating tailored textual summaries from ontologies. In *2nd European Semantic Web Conf.*, pages 531–545, Heraklion, Greece.

J. Clarke and M. Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 1(31):399–429.

H. Dalianis. 1999. Aggregation in natural language generation. *Comput. Intelligence*, 15(4):384–414.

L. Danlos. 1984. Conceptual and linguistic decisions in generation. In *10th COLING*, pages 501–504, Stanford, CA.

S. Demir, S. Carberry, and K.F. McCoy. 2010. A discourse-aware graph-based content-selection framework. In *6th Int. Nat. Lang. Generation Conference*, pages 17–25, Trim, Co. Meath, Ireland.

D. Galanis and I. Androutsopoulos. 2007. Generating multilingual descriptions from linguistically annotated OWL ontologies: the NaturalOWL system. In *11th European Workshop on Natural Lang. Generation*, pages 143–146, Schloss Dagstuhl, Germany.

D. Galanis, G. Karakatsiotis, G. Lampouras, and I. Androutsopoulos. 2009. An open-source natural language generator for OWL ontologies and its use in Protégé and Second Life. In *12th Conf. of the European Chapter of ACL (demos)*, Athens, Greece.

D. Galanis, G. Lampouras, and I. Androutsopoulos. 2012. Extractive multi-document summarization with ILP and Support Vector Regression. In *COLING*, pages 911–926, Mumbai, India.

B.C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler. 2008. OWL 2: The next step for OWL. *Web Semantics*, 6:309–322.

I. Konstas and M. Lapata. 2012a. Concept-to-text generation via discriminative reranking. In *50th Annual Meeting of ACL*, pages 369–378, Jeju Island, Korea.

I. Konstas and M. Lapata. 2012b. Unsupervised concept-to-text generation with hypergraphs. In *HLT-NAACL*, pages 752–761, Montréal, Canada.

P. Kuznetsova, V. Ordonez, A. Berg, T. Berg, and Y. Choi. 2012. Collective generation of natural image descriptions. In *50th Annual Meeting of ACL*, pages 359–368, Jeju Island, Korea.

P. Liang, M. Jordan, and D. Klein. 2009. Learning semantic correspondences with less supervision. In *47th Meeting of ACL and 4th AFNLP*, pages 91–99, Suntec, Singapore.

S.F. Liang, R. Stevens, D. Scott, and A. Rector. 2011. Automatic verbalisation of SNOMED classes using OntoVerbal. In *13th Conf. AI in Medicine*, pages 338–342, Bled, Slovenia.

T. Marciniak and M. Strube. 2005. Beyond the pipeline: Discrete optimization in NLP. In *9th Conference on Computational Natural Language Learning*, pages 136–143, Ann Arbor, MI.

R. McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *European Conference on Information Retrieval*, pages 557–564, Rome, Italy.

C. Mellish and J.Z. Pan. 2008. Natural language directed inference from ontologies. *Artificial Intelligence*, 172:1285–1315.

C. Mellish and X. Sun. 2006. The Semantic Web as a linguistic resource: opportunities for nat. lang. generation. *Knowledge Based Systems*, 19:298–303.

E. Reiter and R. Dale. 2000. *Building Natural Language Generation Systems*. Cambridge Univ. Press.

R. Schwitter, K. Kaljurand, A. Cregan, C. Dolbear, and G. Hart. 2008. A comparison of three controlled nat. languages for OWL 1.1. In *4th OWL Experiences and Directions Workshop*, Washington DC.

R. Schwitter. 2010. Controlled natural languages for knowledge representation. In *23rd COLING*, pages 1113–1121, Beijing, China.

N. Shadbolt, T. Berners-Lee, and W. Hall. 2006. The Semantic Web revisited. *IEEE Intell. Systems*, 21:96–101.

M.A. Walker, O. Rambow, and M. Rogati. 2001. Spot: A trainable sentence planner. In *2nd Annual Meeting of NAACL*, pages 17–24, Pittsburgh, PA.

S. Williams, A. Third, and R. Power. 2011. Levels of organization in ontology verbalization. In *13th European Workshop on Natural Lang. Generation*, pages 158–163, Nancy, France.

K. Woodsend and M. Lapata. 2012. Multiple aspect summarization using ILP. In *EMNLP-CoNLL*, pages 233–243, Jesu Island, Korea.