# Using Integer Linear Programming in Concept-to-Text Generation to Produce More Compact Texts

**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 ILP model of concept-to-text generation. Unlike pipeline architectures, our model jointly considers the choices in content selection, lexicalization, and aggregation to avoid greedy decisions and produce more compact texts.

## 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 (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) 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 to express. The next stage, text planning, ranges from simply ordering the selected facts to 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 then combines sentences into 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, content selection and lexicalization may lead to more or fewer sentence aggregation opportunities.

We present an Integer Linear Programming (ILP) model that combines content selection, lexicalization, and sentence aggregation. Our model does not consider 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 a 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. Given an individual (entity) or class of an OWL ontology and a set of facts (OWL axioms) about the individual or class, we aim to produce a text that expresses as many of the facts in as few words as possible. This is important when space is limited or expensive (e.g., product descriptions on smartphones, advertisements in search engines).

Although the search space of our model is very large and ILP problems are in general NP-hard, ILP solvers can be used, they are very fast in practice, and they guarantee finding a global optimum. Experiments show that our ILP model outperforms, in terms of compression, an NLG system that uses the same components, but connected in a pipeline, with no deterioration in fluency and clarity.

## 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 the importance of each



*Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 561–566,
Sofia, Bulgaria, August 4-9 2013. 

fact or pair of facts, they select the facts to express by formulating an optimization problem similar to energy minimization. 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 that are placed in the same subset are aggregated to form a longer sentence. An ILP model is used to find the partitioning that maximizes the pairwise similarity of the facts in each subset, subject to constraints limiting the number of subsets and the facts in each subset.

Althaus et al. (2004) show that ordering a set of sentences to maximize sentence-to-sentence coherence is equivalent to the traveling salesman problem and, hence, NP-complete. They also show how an ILP solver can be used in practice.

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.

To the best of our knowledge, this article 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. It is also the first article to consider ILP in NLG from OWL ontologies.

## 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 $O$. Hence, we can assume that each fact $f_i$ is a triple $\langle S_i, R_i, O_i\rangle$.[1]

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; for example, sentences with the same verb are easier to aggregate. We use aggregation rules (Dalianis, 1999) that operate on sentence plans and usually lead to shorter texts.

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 $s_j$.[2] 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 $s_j$

[1] 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.

[2] 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.

are counted only once, because they will probably be expressed only once, due to aggregation.

Our objective function (4) maximizes the number of selected facts $f_i$ 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 (sums) 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 $|B|$. In the first part of the objective, we treat all the facts as equally important; if importance scores are also available for the facts, they can be added as multipliers of $\alpha_i$. The parameters $\lambda_1$ and $\lambda_2$ are used to tune the priority given to expressing many facts vs. generating shorter texts; we set $\lambda_1 + \lambda_2 = 1$.

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

subject to:

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

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

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

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

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

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 $\sigma$. 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.[3] We modified content selection, lexicalization, and aggregation to use our ILP model, maintaining the aggregation rules of the original system.[4] For referring expression generation and surface realization, the new system, called ILPNLG, invokes the corresponding components of NaturalOWL.

The original system, 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).

We used the Wine Ontology, which had been

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

[4] We use the Branch and Cut implementation of GLPK; see `sourceforge.net/projects/winglpk/`.

used in previous experiments with PIPELINE.[5] We kept the 2 topical sections, the ordering of sections and relations, and the sentence plans that had been used in the previous experiments, but we added more sentence plans to ensure that 3 sentence plans were available per fact. We generated texts for the 52 wine individuals of the ontology; we did not experiment with texts describing classes of wines, because we could not think of multiple alternative sentence plans for many of their axioms. For each individual, there were 5 facts on average and a maximum of 6 facts.

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, where PIPELINE was allowed to aggregate up to 3 original sentences.

With ILPNLG, we repeated the generation of the texts of the 52 individuals using different values of $\lambda_1$ ($\lambda_2 = 1 - \lambda_1$), 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 observed 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,

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

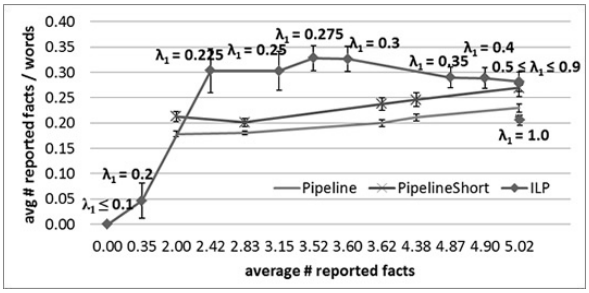


Figure 1: Facts/words ratio of the generated texts.

since 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 $\lambda_1$, 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 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. In all the experiments, the ILP solver was very fast (average: 0.08 sec, worst: 0.14 sec). Experiments with human judges also showed that the texts of ILPNLG cannot be distinguished from those of PIPELINESHORT in terms of fluency and text clarity. Hence, the highest compactness of the texts of ILPNLG does *not* come at the expense of lower text quality. Space does not permit a more detailed description of these experiments.

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.

## 5 Conclusions

We presented an ILP model for NLG that jointly considers the choices in content selection, lexicalization, and aggregation to avoid greedy local decisions and produce more compact texts. Experiments verified that our model can express more facts per word, compared to a pipeline, which is important when space is scarce. An off-the-shelf ILP solver took approximately 0.1 sec for each text. We plan to extend our model to include text planning and referring expressions generation.

## 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.

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 Annual Meeting of ACL*, pages 481–490, Portland, OR.

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.

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 Integer Linear Programming 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. 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.

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 integer linear programming. In *EMNLP-CoNLL*, pages 233–243, Jesu Island, Korea.