Open Source Software como Herramienta en la Reproducibilidad de la Investigación

Danilo Dominguez Perez, PhD.

Acerca de mí

- Danilo Domínguez Pérez
- Estudios
  - Egresado de la UTP - Ing. en Sistemas y Computación
  - Maestría en Ciencias Computacionales del Rochester Institute of Technology
  - Doctorado en Ciencias Computacionales de Iowa State University
    - Investigación en Análisis y Testing de Aplicaciones Móviles
-
- Miembro de FLOSSpa
- Ingeniero de Software Móvil en Automattic

https://www.linkedin.com/in/danilo-dominguez-perez

@danilo04

@danilo04

Investigación Científica

- Proceso creativo y sistemático
- Ayuda a aumentar el conocimiento actual
- Implica recopilación, organización y análisis de información

https://fusemachines.medium.com/how-artificial-intelligence-will-impact-scientific-research-4e6f4face1ae

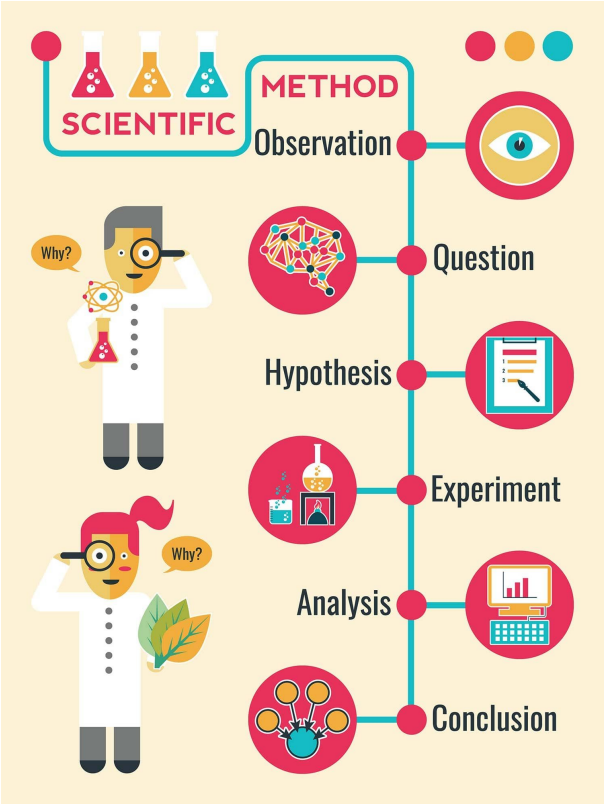
Investigación Científica

- Proceso creativo y sistemático
- Ayuda a aumentar el conocimiento actual
- Implica recopilación, organización y análisis de información
- Aplicación del método científico
- Diferentes tipo de investigación
  - Tecnológica
  - Cultural
  - etc.

https://fusemachines.medium.com/how-artificial-intelligence-will-impact-scientific-research-4e6f4face1ae

Proceso Científico

- Los logros de la investigación se pueden ver palpables desde el Internet, GPS, vacunas, etc.
- Todo este logro es acumulativo y depende de un tejido de confianza en el proceso científico



Proceso Científico

¿Cómo nos aseguramos que los resultados de los experimentos son correctos?

Reproducibilidad en la Investigación Científica

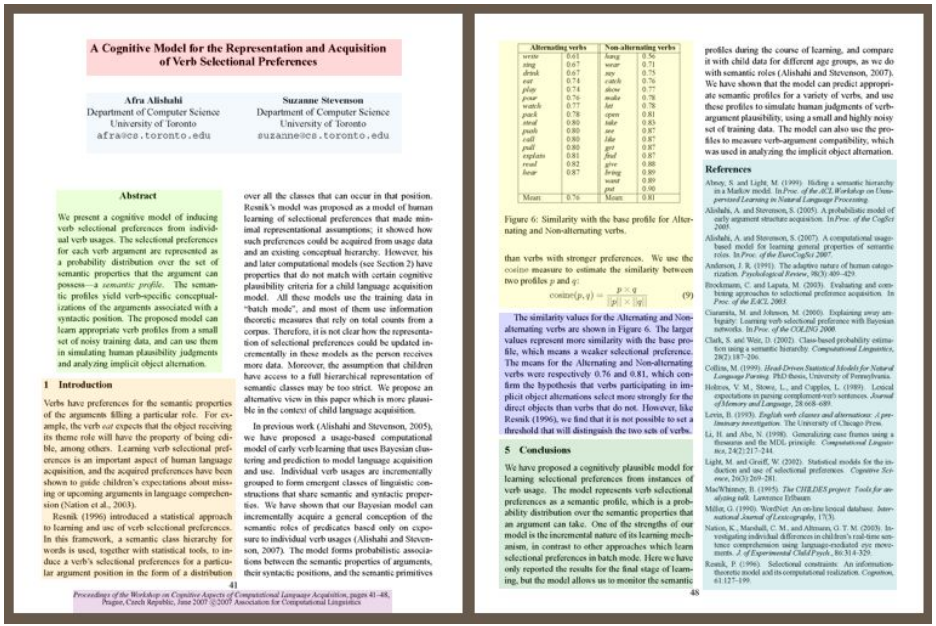
- Imaginense darle 1 receta a 10 chefs y obtener 10 platos diferentes
- Hay diferentes factores y variables que contribuyen a la inconsistencia
- Lo mismo aplica para experimentos científicos

Una manera que se utiliza para validar estudios científicos es repetir la investigación que lo produjo

https://www.nationalacademies.org/news/2019/09/reproducibility-and-replicability-in-research

El Artículo Científico

- Se utilizan para compartir su propio trabajo de investigación original con otros científicos o para revisar la investigación realizada por otros
- Suelen publicarse en un periódico llamado journal cuyo propósito es publicar este tipo de trabajos
- Normalmente presentan el método de experimentación completo, los resultados y el análisis de los mismos
- **Fuente principal para entender y reproducir estudios**



A Cognitive Model for the Representation and Acquisition of Verb Selectional Preferences

Afra Alishahi
Department of Computer Science
University of Toronto
afra@cs.toronto.edu

Suzanne Stevenson
Department of Computer Science
University of Toronto
suzanne@cs.toronto.edu

Abstract

We present a cognitive model of inducing verb selectional preferences from individual verb usages. The selectional preferences for each verb argument are represented as a probability distribution over the set of semantic properties that the argument can possess—a *semantic profile*. The semantic profiles yield verb-specific conceptualizations of the arguments associated with a syntactic position. The proposed model can learn appropriate verb profiles from a small set of noisy training data, and can use them in simulating human plausibility judgments and analyzing implicit object alternation.

1 Introduction

Verbs have preferences for the semantic properties of the arguments filling a particular role. For example, the verb *eat* expects that the object receiving its theme role will have the property of being edible, among others. Learning verb selectional preferences is an important aspect of human language acquisition, and the acquired preferences have been shown to guide children's expectations about missing or upcoming arguments in language comprehension (Nation et al., 2003).

Resnik (1996) introduced a statistical approach to learning and use of verb selectional preferences. In this framework, a semantic class hierarchy for words is used, together with statistical tools, to induce a verb's selectional preferences for a particular argument position in the form of a distribution over all the classes that can occur in that position. Resnik's model was proposed as a model of human learning of selectional preferences that made minimal representational assumptions; it showed how such preferences could be acquired from usage data and an existing conceptual hierarchy. However, his and later computational models (see Section 2) have properties that do not match with certain cognitive plausibility criteria for a child language acquisition model. All these models use the training data in "batch mode", and most of them use information theoretic measures that rely on total counts from a corpus. Therefore, it is not clear how the representation of selectional preferences could be updated incrementally in these models as the person receives more data. Moreover, the assumption that children have access to a full hierarchical representation of semantic classes may be too strict. We propose an alternative view in this paper which is more plausible in the context of child language acquisition.

In previous work (Alishahi and Stevenson, 2005), we have proposed a usage-based computational model of early verb learning that uses Bayesian clustering and prediction to model language acquisition and use. Individual verb usages are incrementally grouped to form emergent classes of linguistic constructions that share semantic and syntactic properties. We have shown that our Bayesian model can incrementally acquire a general conception of the semantic roles of predicates based only on exposure to individual verb usages (Alishahi and Stevenson, 2007). The model forms probabilistic associations between the semantic properties of arguments, their syntactic positions, and the semantic primitives

41

Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition, pages 41–48,
Prague, Czech Republic, June 2007 ©2007 Association for Computational Linguistics

Alternating verbs		Non-alternating verbs	
write	0.61	*bang*	0.56
sing	0.67	*wear*	0.71
drink	0.67	*say*	0.75
eat	0.74	*catch*	0.76
play	0.74	*show*	0.77
pour	0.76	*make*	0.78
watch	0.77	*hit*	0.78
pack	0.78	*open*	0.81
steal	0.80	*take*	0.83
push	0.80	*see*	0.87
call	0.80	*like*	0.87
pull	0.80	*get*	0.87
explain	0.81	*find*	0.87
read	0.82	*give*	0.88
hear	0.87	*bring*	0.89
		want	0.89
		put	0.90
Mean	0.76	Mean	0.81

Figure 6: Similarity with the base profile for Alternating and Non-alternating verbs.

than verbs with stronger preferences. We use the cosine measure to estimate the similarity between two profiles p and q:

$$\text{cosine}(p,q) = \frac{p \times q}{|p| \times |q|} \quad (9)$$

The similarity values for the Alternating and Non-alternating verbs are shown in Figure 6. The larger values represent more similarity with the base profile, which means a weaker selectional preference. The means for the Alternating and Non-alternating verbs were respectively 0.76 and 0.81, which confirm the hypothesis that verbs participating in implicit object alternations select more strongly for the direct objects than verbs that do not. However, like Resnik (1996), we find that it is not possible to set a threshold that will distinguish the two sets of verbs.

5 Conclusions

We have proposed a cognitively plausible model for learning selectional preferences from instances of verb usage. The model represents verb selectional preferences as a semantic profile, which is a probability distribution over the semantic properties that an argument can take. One of the strengths of our model is the incremental nature of its learning mechanism, in contrast to other approaches which learn selectional preferences in batch mode. Here we have only reported the results for the final stage of learning, but the model allows us to monitor the semantic profiles during the course of learning, and compare it with child data for different age groups, as we do with semantic roles (Alishahi and Stevenson, 2007). We have shown that the model can predict appropriate semantic profiles for a variety of verbs, and use these profiles to simulate human judgments of verb-argument plausibility, using a small and highly noisy set of training data. The model can also use the profiles to measure verb-argument compatibility, which was used in analyzing the implicit object alternation.

References

Abney, S. and Light, M. (1999). Hiding a semantic hierarchy in a Markov model. In *Proc. of the ACL Workshop on Unsupervised Learning in Natural Language Processing*.

Alishahi, A. and Stevenson, S. (2005). A probabilistic model of early argument structure acquisition. In *Proc. of the CogSci 2005*.

Alishahi, A. and Stevenson, S. (2007). A computational usage-based model for learning general properties of semantic roles. In *Proc. of the EuroCogSci 2007*.

Anderson, J. R. (1991). The adaptive nature of human categorization. *Psychological Review*, 98(3):409–429.

Brockmann, C. and Lapata, M. (2003). Evaluating and combining approaches to selectional preference acquisition. In *Proc. of the EACL 2003*.

Ciaramita, M. and Johnson, M. (2000). Explaining away ambiguity: Learning verb selectional preference with Bayesian networks. In *Proc. of the COLING 2000*.

Clark, S. and Weir, D. (2002). Class-based probability estimation using a semantic hierarchy. *Computational Linguistics*, 28(2):187–206.

Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania.

Holmes, V. M., Stowe, L., and Cupples, L. (1989). Lexical expectations in parsing complement-verb sentences. *Journal of Memory and Language*, 28:668–689.

Levin, B. (1993). *English verb classes and alternations: A preliminary investigation*. The University of Chicago Press.

Li, H. and Abe, N. (1998). Generalizing case frames using a thesaurus and the MDL principle. *Computational Linguistics*, 24(2):217–244.

Light, M. and Greiff, W. (2002). Statistical models for the induction and use of selectional preferences. *Cognitive Science*, 26(3):269–281.

MacWhinney, B. (1995). *The CHILDES project: Tools for analyzing talk*. Lawrence Erlbaum.

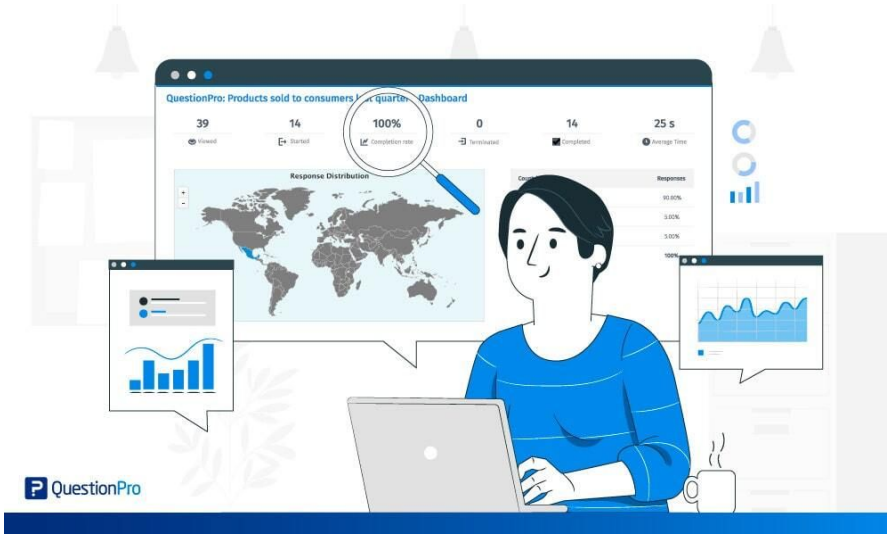
Miller, G. (1990). WordNet: An on-line lexical database. *International Journal of Lexicography*, 17(3).

Nation, K., Marshall, C. M., and Altmann, G. T. M. (2003). Investigating individual differences in children's real-time sentence comprehension using language-mediated eye movements. *J. of Experimental Child Psych.*, 86:314–329.

Resnik, P. (1996). Selectional constraints: An information-theoretic model and its computational realization. *Cognition*, 61:127–159.

48

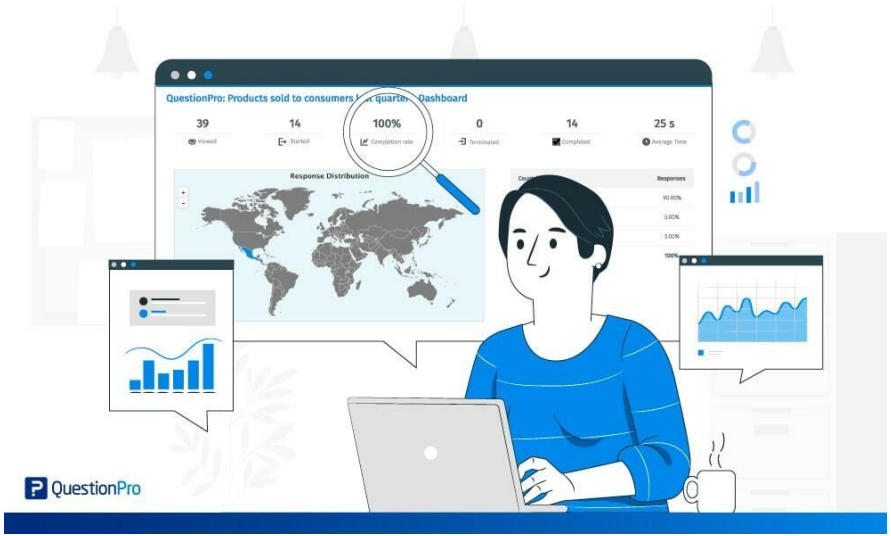
Organización y Análisis de Información



https://www.questionpro.com/blog/what-is-data-analysis/

- Anteriormente era un proceso manual y tedioso
- Estaba restringido a la capacidad humana de procesar los datos
- Por ende, el artículo científico era suficiente para validar experimentos
- Las computadoras se han convertido en un vehículo para ampliar las capacidades de análisis humanas

Organización y Análisis de Información



- Utilizamos programas computacionales para organizar, limpiar y analizar información
- Estos programas utilizan diferentes algoritmos, análisis estadístico para procesar datos
- Pueden contener **bugs**
- Los datos pueden estar **corruptos**

https://www.questionpro.com/blog/what-is-data-analysis/

“La informática es ahora una actividad cotidiana para todos los científicos, de una forma u otra, desde ejecutar estadísticas simples sobre datos empíricos hasta ejecutar simulaciones masivas en instalaciones informáticas de liderazgo.”

Barba (2022)

¿Cómo nos aseguramos que nuestros experimentos son **reproducibles**?

Definición de Reproducibilidad

“Obtener resultados computacionales consistentes usando los mismos datos de entrada, pasos computacionales, métodos y código, y condiciones de análisis.”

National Academies of Sciences, Engineering, and Medicine (2019)

Implicaciones

- En 2011, ESEC/FSE inició un experimento novedoso para una importante conferencia de software: dar a los autores la oportunidad de enviar para evaluación cualquier artefacto que acompañe a sus artículos
- Un experimento similar se ha llevado a cabo con éxito en varias conferencias más.
- Este documento describe los objetivos y la mecánica general de este proceso

Implicaciones

- La definición indica que los datos y el código producido por investigadores deben de estar disponibles para otros investigadores
- No necesariamente se deben liberar con licencias Open Source
- Al no liberar los datos y código de los experimentos, otros investigadores no tienen muchas libertades sobre los datos
- Puede ser un proceso tedioso contactar a los autores para que provean la información
- ¿Qué pasa si son laboratorios rivales?

Barba (2022)

Defining the role of open source software in research reproducibility

Lorena A. Barba, the George Washington University, Washington D.C.

November 2021

Abstract

Reproducibility is inseparable from transparency, as sharing data, code and computational environment is a pre-requisite for being able to retrace the steps of producing the research results. Others have made the case that this artifact sharing should adopt appropriate licensing schemes that permit reuse, modification and redistribution. I make a new proposal for the role of open source software, stemming from the lessons it teaches about distributed collaboration and a commitment-based culture. Reviewing the defining features of open source software (licensing, development, communities), I look for explanation of its success from the perspectives of connectivism—a learning theory for the digital age—and the language-action framework of Winograd and Flores. I contend that reproducibility engenders trust, which we routinely build in community via conversations, and the practices of open source software help us to learn how to be more effective learning (discovering) together, contributing to the same goal.

Historia de FOSS

- Investigadores han estado involucrados desde la génesis del software de código abierto (OSS) hace unos 50 años
- Ejemplos
  - Unix en Bell Labs
  - Berkeley Software Distribution
  - Unix-compatible GNU system en el MIT
  - Linux por Linus Torvalds en la Universidad de Helsinki
- El término software de código abierto (open source software) se introdujo hace unos 24 años (Peterson, 2008)
- Open Source Initiative (OSI) estipula el cumplimiento de diez criterios, que incluyen no solo la disponibilidad del código fuente, sino también la vinculación de una licencia para usarlo, modificarlo y redistribuirlo libremente.

Implicaciones de Open Source

- Simplemente ofreciendo el código no es suficiente
- Se debe incluir una licencia aprobada por el OSI
- Para todo trabajo creativo **copyright** es automáticamente asignado a tu código
- El beneficio clave de las licencias aprobadas por OSI es que los investigadores no necesitan una amplia capacitación legal o consultores para navegar estos problemas: las licencias están “**preempaquetadas**” y listas para usar
- Cuando se aplica al software de investigación, contribuye a la transparencia del flujo de trabajo computacional y la disponibilidad para que otros usen/modifiquen/redistribuyan el software, uno de los requisitos de la **reproducibilidad**

Implicaciones de Open Source

- El desarrollo abierto conduce a una mejor calidad porque los usuarios pueden generar informes de errores conscientes de la fuente y cooperar con los desarrolladores
- Muchos grupos de investigación pueden trabajar en mejorar el proyecto
- Los investigadores no tienen que estar firmando NDAs ni nada por el estilo

La Ciencia es una Conversación

- Conocimiento científico no solo se transfiere por artículos científicos
- Muchas ideas nacen de conversaciones entre investigadores
- La apertura (Openness) promueve redes ricas, comunidades animadas y conexiones fértiles. Y esto es bueno para la ciencia
- Los proyectos de software de código abierto y la cultura de sus comunidades tienen más que ofrecer que un esquema para garantizar la libertad de las restricciones de derechos de autor en el código compartido
- Las comunidades de código abierto han desarrollado plantillas para coordinar las acciones de diversos grupos de personas, con el objetivo de mejorar la comunicación y trabajar juntos de manera más efectiva.

Open Source + Investigación

LLVM

- LLVM es un conjunto de tecnologías de compilador y cadena de herramientas
- Desarrollado en University of Illinois at Urbana–Champaign
- Por Chris Lattner y su tutor Vikram Adve
- Licencia UIUC (BSD-style), basada MIT/X11 y licencia BSD de 3 cláusulas
- Es la base de múltiples proyectos
  - Rust
  - Clang
  - Swift
  - Xcode
- Se maneja a través de una fundación: https://foundation.llvm.org/

Compilador Soot

- Soot comenzó como un marco de optimización de Java.
- Actualmente se utiliza para analizar, instrumentar, optimizar y visualizar aplicaciones Java y Android.
- Licencia GNU Lesser General Public License v2.1
- Permite análisis estático de programas Java y Android

Project Jupyter

- Creado a partir de IPython en 2014 por Fernando Pérez
  - Colombiano, Profesor de Estadística en la Universidad de UC Berkeley
- Se separó de IPython en 2014
- Se maneja a través de una fundación: https://jupyter.org/about
- IPython tiene licencia BSD

¿Preguntas?